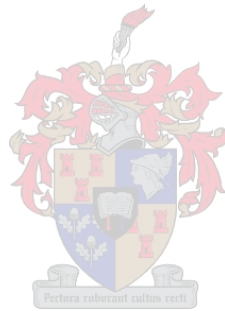


Edge Criticality in Secure Graph Domination



Anton Pierre de Villiers

Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Science at Stellenbosch University

Promoter: Prof JH van Vuuren
Co-promoter: Dr AP Burger

December 2014

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 1, 2014

Abstract

The domination number of a graph is the cardinality of a smallest subset of its vertex set with the property that each vertex of the graph is in the subset or adjacent to a vertex in the subset. This graph parameter has been studied extensively since its introduction during the early 1960s and finds application in the generic setting where the vertices of the graph denote physical entities that are typically geographically dispersed and have to be monitored efficiently, while the graph edges model links between these entities which enable guards, stationed at the vertices, to *monitor* adjacent entities.

In the above application, the guards remain stationary at the entities. In 2005, this constraint was, however, relaxed by the introduction of a new domination-related parameter, called the *secure domination number*. In this relaxed, dynamic setting, each unoccupied entity is *defended* by a guard stationed at an adjacent entity who can travel along an edge to the unoccupied entity in order to resolve a security threat that may occur there, after which the resulting configuration of guards at the entities is again required to be a dominating set of the graph. The secure domination number of a graph is the smallest number of guards that can be placed on its vertices so as to satisfy these requirements.

In this generalised setting, the notion of edge removal is important, because one might seek the cost, in terms of the additional number of guards required, of protecting the complex of entities modelled by the graph if a number of edges in the graph were to fail (*i.e.* a number of links were to be eliminated from the complex, thereby disqualifying guards from moving along such disabled links).

A comprehensive survey of the literature on secure graph domination is conducted in this dissertation. Descriptions of related, generalised graph protection parameters are also given. The classes of graphs with secure domination number 1, 2 or 3 are characterised and a result on the number of defenders in any minimum secure dominating set of a graph without end-vertices is presented, after which it is shown that the decision problem associated with computing the secure domination number of an arbitrary graph is **NP-complete**.

Two exponential-time algorithms and a binary programming problem formulation are presented for computing the secure domination number of an arbitrary graph, while a linear algorithm is put forward for computing the secure domination number of an arbitrary tree. The practical efficiencies of these algorithms are compared in the context of small graphs.

The smallest and largest increase in the secure domination number of a graph are also considered when a fixed number of edges are removed from the graph. Two novel cost functions are introduced for this purpose. General bounds on these two cost functions are established, and exact values of or tighter bounds on the cost functions are determined for various infinite classes of special graphs.

Threshold information is finally established in respect of the number of possible edge removals from a graph before increasing its secure domination number. The notions of *criticality* and *stability* are introduced and studied in this respect, focussing on the smallest number of arbitrary edges whose deletion *necessarily* increases the secure domination number of the resulting graph, and the largest number of arbitrary edges whose deletion *necessarily does not* increase the secure domination number of the resulting graph.

Uittreksel

Die dominasiegetal van 'n grafiek is die kardinaalgetal van 'n kleinste deelversameling van die grafiek se puntversameling met die eienskap dat elke punt van die grafiek in die deelversameling is of naasliggend is aan 'n punt in die deelversameling. Hierdie grafiekparameter is sedert die vroeë 1960s uitvoerig bestudeer en vind toepassing in die generiese situasie waar die punte van die grafiek fisiese entiteite voorstel wat tipies geografies verspreid is en doeltreffend gemonitor moet word, terwyl die lyne van die grafiek skakels tussen hierdie entiteite voorstel waarlangs wagne, wat by die entiteite gebaseer is, naasliggende entiteite kan *monitor*.

In die bogenoemde toepassing, bly die wagne bewegingloos by die fisiese entiteite waar hulle geplaas word. In 2005 is hierdie beperking egter verslap met die daarstelling van 'n nuwe dominasie-verwante grafiekparameter, bekend as die *sekre dominasiegetal*. In hierdie verslappede, dinamiese situasie word elke punt sonder 'n wagne deur 'n wagne *verdedig* wat by 'n naasliggende punt geplaas is en wat langs die verbindingslyn na die leë punt kan beweeg om daar 'n bedreiging te neutraliseer, waarna die gevolglike plasing van wagne weer 'n dominasieversameling van die grafiek moet vorm. Die sekre dominasiegetal van 'n grafiek is die kleinste getal wagne wat op die punte van die grafiek geplaas kan word om aan hierdie vereistes te voldoen.

Die beginsel van lynverwydering speel 'n belangrike rol in hierdie veralgemeende situasie, omdat daar gevra mag word na die koste, in terme van die addisionele getal wagne wat vereis word, om die kompleks van entiteite wat deur die grafiek gemodelleer word, te beveilig indien 'n aantal lynfalings in die grafiek plaasvind (*m.a.w.* indien 'n aantal skakels uit die kompleks van entiteite verwyder word, en wagne dus nie meer langs sulke skakels mag beweeg nie).

'n Omvattende literatuurstudie oor sekre dominasie van grafieke word in hierdie verhandeling gedoen. Beskrywings van verwante, veralgemeende verdedigingsparameters in grafiekteorie word ook gegee. Die klasse van grafieke met sekre dominasiegetal 1, 2 of 3 word gekarakteriseer en 'n resultaat oor die getal verdedigers in enige kleinste sekre dominasieversameling van 'n grafiek sonder endpunte word daargestel, waarna daar getoon word dat die beslissingsprobleem onderliggend aan die berekening van die sekre dominasiegetal van 'n arbitrêre grafiek **NP-volledig** is.

Twee eksponensiële-tyd algoritmes en 'n binêre programmeringsformulering word vir die bepaling van die sekre dominasiegetal van 'n arbitrêre grafiek daargestel, terwyl 'n lineêre algoritme vir die berekening van die sekre dominasiegetal van 'n arbitrêre boom ontwerp word. Die praktiese doeltreffendhede van hierdie algoritmes word vir klein grafieke met mekaar vergelyk.

Die kleinste en grootste toename in die sekre dominasiegetal van 'n grafiek word ook oorweeg wanneer 'n vaste getal lyne uit die grafiek verwyder word. Twee nuwe kostefunksies word vir hierdie doel daargestel en algemene grense word op hierdie kostefunksies vir arbitrêre grafieke bepaal, terwyl eksakte waardes van of verbeterde grense op hierdie kostefunksies vir verskeie oneindige klasse van spesiale grafieke bereken word.

Drempelinligting word uiteindelik bepaal in terme van die moontlike getal lynverwyderings uit 'n grafiek voordat die sekre dominasiegetal daarvan toeneem. Die konsepte van *kritiekheid* en *stabiliteit* word in hierdie konteks bestudeer, met 'n fokus op die kleinste getal arbitrêre lynfalings wat *noodwendig* die sekre dominasiegetal van die gevolglike grafiek laat toeneem, of die grootste getal arbitrêre lynfalings wat *noodwendig* die sekre dominasiegetal van die gevolglike grafiek onveranderd laat.

Acknowledgements

The author wishes to acknowledge the following people for their various contributions towards the completion of this work:

- I wish to extend my deepest gratitude to my promoter, Prof Jan van Vuuren, for his excellent guidance throughout this project. I appreciate his time, patience and enthusiasm when listening to my ideas, as well as his dedication and hard work in ensuring that work of a high standard is delivered.
- I wish also to extend my deepest gratitude to my co-promoter, Dr Alewyn Burger, for his undying enthusiasm and accessibility, friendliness and his willingness to lend a helping hand whenever necessary.
- I wish to thank the Department of Industrial Engineering for the use of their computing facilities and office space.
- I wish to thank the National Research Foundation for their financial support in the form of a bursary (under grant number 81558) during the past three years.
- A big thanks also goes to the various masters and doctoral students with whom I shared a number of wonderful experiences throughout the past three years.
- Finally, I extend my deepest thanks to my family and friends for their unwavering support during the past three years.

Table of Contents

List of Reserved Symbols	xiii
List of Figures	xvii
List of Tables	xxi
List of Algorithms	xxiii
1 Introduction	1
1.1 Dominating and defending chessboards	1
1.2 Casting the net wider: Informal problem description	7
1.3 Dissertation scope and objectives	10
1.4 Dissertation organisation	11
2 Mathematical preliminaries	13
2.1 Basic notions in graph theory	13
2.1.1 Neighbourhoods	14
2.1.2 Graph isomorphisms and subgraphs	15
2.1.3 Connected Graphs	16
2.1.4 Graph operations and special graphs	20
2.2 Fundamentals of graph domination	24
2.2.1 Domination, independence and irredundance	24
2.2.2 The domination number of a graph	26
2.2.3 Multiple domination	30
2.2.4 Edge criticality in graph domination	30
2.3 Basic notions of complexity theory	33
2.3.1 The complexity classes P and NP	34
2.3.2 Reductions and NP -completeness	35
2.3.3 The satisfiability decision problem	36

2.3.4	From decision problems to computation problems	37
2.3.5	Graph domination is NP-complete	38
2.3.6	Algorithms for computing the domination number of a graph	39
2.4	Chapter summary	44
3	Literature review	47
3.1	Basic notions of graph protection	47
3.2	Secure graph domination	52
3.2.1	Bounds on γ_s	52
3.2.2	Special graph classes	54
3.2.3	Edge removal criticality in secure graph domination	56
3.3	Generalised graph protection parameters	59
3.3.1	Finite higher order generalisations	59
3.3.2	Generalisations to include infinitely many moves	61
3.3.3	Generalisations involving multiple moves	61
3.4	Chapter summary	62
4	Properties of minimum secure dominating sets	65
4.1	The structure of a secure dominating set	65
4.2	Graphs with small secure domination numbers	66
4.3	On the defenders in a minimum secure dominating set	72
4.4	Secure domination is NP-complete	75
4.5	Chapter summary	77
5	Algorithms for secure domination	79
5.1	Computing the secure domination number of an arbitrary graph	79
5.1.1	A branch-and-reduce algorithm	81
5.1.2	A branch-and-bound algorithm	84
5.1.3	Worst-case complexity analysis	87
5.1.4	Numerical results	87
5.1.5	A binary programming problem formulation	90
5.2	Computing the secure domination number of a tree	92
5.2.1	Spiders	92
5.2.2	From spiders to trees in general	97
5.2.3	Linear algorithmic implementation	102
5.2.4	Validation of the algorithm	107

5.3	Chapter summary	107
6	Edge failure and secure graph domination	111
6.1	Two functions measuring the cost of edge failures	111
6.2	General bounds on the cost functions c_q and C_q	113
6.3	The cost functions c_q and C_q for paths and cycles	117
6.4	The cost functions c_q and C_q for wheels	122
6.5	The cost functions c_q and C_q for complete bipartite graphs	126
6.6	The cost functions c_q and C_q for complete graphs	131
6.7	Chapter summary	134
7	Criticality and stability in secure graph domination	137
7.1	The notions of p -stability and q -criticality	137
7.2	Characterisation of q -critical graphs of order n	140
7.3	Computing p -stable graphs of order n	142
7.4	Extremal stability and criticality values	148
7.4.1	The largest q for which an order n graph can be q -critical	148
7.4.2	The largest p for which an order n graph can be p -stable	149
7.5	Special graph classes	159
7.5.1	Complete and complete bipartite graphs	159
7.5.2	Paths and cycles	160
7.5.3	Dumbbells	163
7.6	Chapter summary	164
8	Conclusion	165
8.1	Dissertation summary	165
8.2	Appraisal of dissertation contributions	167
8.3	Future work	168
	References	173
A	Properties of Floor and Ceiling Operations	179
B	Repositories of critical and stable graphs	183
C	Contents of the accompanying compact disc	191

List of Reserved Symbols

Basic graph notation

$G(V, E)$	A graph G with vertex set $V = V(G)$ and edge set $E = E(G)$
$V(G)$	The vertex set of a graph G
$E(G)$	The edge set of a graph G
\mathcal{G}	A family of graphs
$n(G)$	The order of the graph G , <i>i.e.</i> $n(G) = V(G) $
$m(G)$	The size of the graph G , <i>i.e.</i> $m(G) = E(G) $
$N_G(v)$	The open neighbourhood of a vertex v in a graph G
$N_G[v]$	The closed neighbourhood of a vertex v in a graph G
$N_S(v)$	The open neighbourhood of a vertex subset $S \subseteq V(G)$ in a graph G
$N_S[v]$	The closed neighbourhood of a vertex subset $S \subseteq V(G)$ in a graph G
$d_G(u, v)$	The distance between two vertices u and v in a graph G
$\deg_G(v)$	The degree of a vertex v in a graph G
$\delta(G)$	The minimum vertex degree of a graph G
$\Delta(G)$	The maximum vertex degree of a graph G
$\text{pn}(v, S)$	The private neighbourhood set of v with respect to S
$\text{Epn}(v, S)$	The external private neighbourhood set of v with respect to S

Graph relations and operations

$\phi : V(G) \rightarrow V(H)$	An isomorphism from a graph G to a graph H
$G \cong H$	Two graphs G and H are isomorphic
\bar{G}	The complement of a graph G
$H \subseteq G$	The graph H is a subgraph of the graph G
$H \subset G$	The graph H is a proper subgraph of the graph G
$\langle S \rangle_G$	The subgraph of a graph G induced by a given subset S of $V(G)$
$G - S$ [$G - J$]	The subgraph of a graph G resulting from the deletion of a vertex subset S [an edge subset J] from the graph G
$G - v$ [$G - e$]	The subgraph of a graph G resulting from the deletion of a single vertex v [an edge e] from the graph G
$G + J$ [$G + e$]	The subgraph of a graph G resulting from the addition of an edge subset J [an edge e] of $E(G)$ to the graph G
$G - qe$	The class of non-isomorphic graphs obtained by the deletion of q edges from the graph G
$G + qe$	The class of non-isomorphic graphs obtained by the addition of q edges to the graph G
$G \cup H$	The union of two graphs G and H

$G + H$	The join of two graphs G and H
$G \square H$	The cartesian product of two graphs G and H
ℓG	The vertex disjoint union of $\ell \geq 1$ isomorphic copies of the graph G

Special types of graphs

Q_n	The queen's graph of order $n \times n$.
K_n	The king's graph of order $n \times n$.
B_n	The bishop's graph of order $n \times n$.
\mathcal{N}_n	The knight's graph of order $n \times n$.
\mathcal{R}_n	The rook's graph of order $n \times n$.
\mathcal{P}_n	The path of order n
\mathcal{C}_n	The cycle of order n
\mathcal{K}_n	The complete graph of order n
$\overline{\mathcal{K}_n}$	The empty graph
$\mathcal{K}_{1,n-1}$	The star of order n
$\mathcal{K}_{n_1, \dots, n_k}$	The complete multipartite graph with partite set cardinalities $n_1 \leq \dots \leq n_k$
$\mathcal{K}_{k \times n}$	The complete, balanced multipartite graph with k partite sets, each of cardinality n
$\mathcal{C}_n \langle i_1, \dots, i_z \rangle$	The circulant graph of order n with connection set $\{i_1, \dots, i_z\}$
$\mathcal{H}_{p \times q}$	The $p \times q$ hexagonal graph
\mathcal{W}_n	The wheel graph of order $n + 1$
$\mathcal{S}(M \times N)$	The spider consisting of M pendent paths of order N , each joined as a pendent path to a single vertex
$\mathcal{S}(n_1, n_2, \dots, n_k)$	The wounded spider with pendent paths of orders n_1, n_2, \dots, n_k , joined to a single vertex
$\mathcal{D}_{a,b}$	The dumbbell graph of order $a + b$
$\mathcal{D}_{a,b,c}$	The double dumbbell graph of order $a + b + c$
\mathcal{B}_n	The book graph of order n

Graph parameters and classes

$\gamma(G)$	Domination number of the graph G
$\chi(G)$	The (vertex) chromatic number of a graph G
$\gamma_s(G)$	The secure domination number of a graph G
$\gamma_R(G)$	The Roman domination number of a graph G
$\gamma_r(G)$	The weak Roman domination number of a graph G
$\gamma_t(G)$	The total domination number of a graph G
$\Gamma(G)$	The upper domination number of a graph G
$i(G)$	The independent domination number of a graph G
f	A guard function of a graph
$w(f)$	The weight of a guard function f
$\gamma_{st}(G)$	The secure total domination number of a graph G
$\gamma_{s,k}(G)$	The k -secure domination number of a graph G
$\gamma_{s,\infty}(G)$	The ∞ -secure domination number of a graph G
$\gamma_{\ell,k}(G)$	The k^{th} -order ℓ -domination number of a graph G
$\gamma_{s,m}(G)$	The eternal m -security number of a graph G
$b(G)$	The bondage number of a graph G
$\delta'(G)$	The minimum degree of any edge in a graph G

$\kappa_1(G)$	The edge connectivity of a graph G
$\theta_c(G)$	The minimum colonisation weight of a graph G
$\alpha_m(G)$	The eternal vertex cover number of a graph G
$\alpha(G)$	The vertex cover number of a graph G
$\theta(G)$	The clique covering number of a graph G
\mathcal{Q}_n^q	The class of q -critical graphs of order n
\mathcal{S}_n^p	The class of p -stable graphs of order n
Ω_n	The largest value of q for which a graph of order n is q -critical
ω_n	The largest value of p for which a graph of order n is p -stable

List of Figures

1.1	The possible movements of a queen on an 8×8 chessboard	1
1.2	Queens dominating all the squares on an 8×8 chessboard	2
1.3	Five queens do not defend an 8×8 chessboard	2
1.4	Seven queens defend an 8×8 chessboard	3
1.5	Seven queens defend each square on an 8×8 chessboard	4
1.6	Kings, bishops, knights and rooks defending all the squares on a chessboard . . .	5
1.7	The smallest number of queens defend an $n \times n$ chessboard	5
1.8	The movement of a queen being obstructed an 8×8 chessboard	6
1.9	A 4×4 chessboard, \mathcal{Q}_4 , K_4 , B_4 , \mathcal{N}_4 and \mathcal{R}_4	8
1.10	The cost functions $c_q(\mathcal{R}_3)$ and $C_q(\mathcal{R}_3)$	10
2.1	A graph and its complement	14
2.2	The notions of an isomorphism and of equality between graphs	15
2.3	The notion of a subgraph	16
2.4	The notions of a spanning tree and a spanning forest	17
2.5	The notions of internal vertices, leaves, children and the root in a rooted tree . .	18
2.6	The notions of vertex deletion and edge deletion	18
2.7	The adjacency matrix and the weight matrix of a graph	18
2.8	Path graphs	20
2.9	Cycle graphs	20
2.10	The notions of graph union, graph join and the cartesian product of graphs . . .	20
2.11	The notions of a regular graph and a perfect matching	21
2.12	Examples of complete graphs	21
2.13	The notions of a circulant graph and its factors	22
2.14	Examples of a corona, wheel, spider and wounded spider	23
2.15	The hexagonal graph	23
2.16	The notions of domination and independence	24

2.17	The notion of irredundance	25
2.18	The family \mathcal{A} of forbidden graphs in Theorem 2.9	27
2.19	The complexity classes P , NP and co-NP	35
2.20	The complexity classes P , NP , co-NP , NP-hard and NP-complete	36
2.21	Reduction from an instance of 3-SAT to an instance of the dominating set problem	38
2.22	A tree T_1 and a minimum dominating set of T_1	39
2.23	The branching depth-first search tree for computing a minimum set cover	41
2.24	A rooted tree T_2 together with its parent array	45
3.1	A graph of the eight regions of the Roman Empire during the fourth century A.D.	48
3.2	Various defence strategies of the Roman Empire	50
3.3	A γ_s -set $X = X_1 \cup X_2 \cup X_3 \cup X_4$ in a γ_s -ER-critical graph	57
3.4	All γ_s -ER-critical graphs of order 6	58
3.5	Constructing the γ_s -ER-critical graph $\mathcal{K}_3 \cup \mathcal{K}_2 \cup \mathcal{K}_1$	58
3.6	Constructing the γ_s -ER-critical graph in Figure 3.4(n)	59
4.1	An example of the vertex set partition in (4.1)	66
4.2	The graph $\Phi(i, j, k, \ell)$	67
4.3	A spanning subgraph of G in Case 1 of the proof of Theorem 4.2	67
4.4	A spanning subgraph of G in Case 2 of the proof of Theorem 4.2	68
4.5	A spanning subgraph of G in Case 3 of the proof of Theorem 4.2	68
4.6	The dumbbell graph $\mathcal{D}_{3,4}$	69
4.7	Various infinite graph classes for which $\gamma_s = 2$ together with their certificates . .	69
4.8	The graph $\Psi(i, j, k, r, s, t)$	70
4.9	A spanning subgraph of G in Case 1 of the proof of Theorem 4.3	71
4.10	A spanning subgraph of G in Case 2 of the proof of Theorem 4.3	71
4.11	A spanning subgraph of G in Case 3 of the proof of Theorem 4.3	71
4.12	A spanning subgraph of G in Case 4 of the proof of Theorem 4.3	72
4.13	The double dumbbell graph $\mathcal{D}_{3,5,4}$	72
4.14	Various infinite graph classes for which $\gamma_s = 3$ together with their certificates . .	73
4.15	A minimum secure dominating set containing non-defenders	74
4.16	The secure dominating set in the proof of Theorem 4.4	74
4.17	Minimum secure dominating sets of a graph containing defenders	75
4.18	Pendent paths of length 4	76
5.1	Identification and pruning of end-clusters of a spanning tree	80
5.2	A graph G_{11} with two support vertices and no redundancy classes	82

5.3	The secure dominating set of G_{11} by the branch-and-reduce depth-first search tree	83
5.4	The secure dominating set of G_{11} by the branch-and-bound depth-first search tree	86
5.5	Examples of minimum secure dominating sets of $\mathcal{P}_n \square \mathcal{P}_n$ and $\mathcal{H}_{5,5}$	91
5.6	Dominating sets of paths of various lengths that are not secure dominating sets	92
5.7	The spider $\mathcal{S}(3, 3, 1)$ with anchor b	93
5.8	The unique subset $\Lambda_i(\mathcal{S})$ of the vertex set of an extended leg L_i of a spider \mathcal{S}	93
5.9	The solid vertices included in the set $\Lambda_i(\mathcal{S})$ for different values of a_i^*	94
5.10	The set $\Lambda(\mathcal{S})$ for the spiders $\mathcal{S}(7, 7, 7)$ and $\mathcal{S}(3, 3, 1)$	94
5.11	The set $\bar{\Lambda}(\mathcal{S})$ for the spiders $\mathcal{S}(3, 1, 8, 2)$ and $\mathcal{S}(2, 7, 9)$	97
5.12	A tree T_1 and three pendent spiders of T_1	97
5.13	The label lists of the vertices of the pendent spider \mathcal{S}_1 of the tree T_1 in Figure 5.12(b)	99
5.14	Decision tree of binary questions for determining the effective label $\ell(b)$	99
5.15	Using the effective label in the continuation of the secure dominating set pattern	100
5.16	Forming a tree T'_i by replacing a pendent spider \mathcal{S}_i of T_i by a path $\mathcal{P}_{1+\ell(b_i)}$	102
5.17	Identifying, securely dominating and pruning pendent spiders of a tree	103
5.18	A canonical ordering of the vertices of the tree T_1 in Figure 5.12(a)	105
6.1	A graphical illustration of the bounds on $c_q(G)$ and $C_q(G)$ in Corollary 6.2	114
6.2	A graphical illustration of the bounds on $c_q(G)$ and $C_q(G)$ in Corollary 6.4	115
6.3	A minimum secure dominating set of $G = \mathcal{K}_{2,n-2} - qe$, where $V_x^G = V_y^G = \emptyset$	127
6.4	A minimum secure dominating set of $G = \mathcal{K}_{2,n-2} - qe$, where $V_{xy}^G = \emptyset$	127
6.5	A minimum secure dominating set of $G = \mathcal{K}_{2,n-2} - qe$, where $V_{xy}^G \neq \emptyset$	128
6.6	Examples showing that the bounding sequences in Theorem 6.14 are not exact	130
6.7	A graphical illustration of the bound on $C_q(\mathcal{K}_n)$ in Theorem 6.16	133
6.8	The bounding sequences of Algorithm 6.1 and Theorem 6.16 are not exact	134
7.1	The edge-removal metagraph of the complete graph \mathcal{K}_4 of order 4	139
7.2	The 4-step process for constructing the seven 1-critical graphs of order 5	141
7.3	The graph classes $\mathcal{Q}_5^1, \dots, \mathcal{Q}_5^6$	143
7.4	A subset of the graph class \mathcal{S}_5^0	145
7.5	A minimum secure dominating set $X = \{v_1, v_2\}$ for a graph which is not in \mathcal{S}_5^0	146
7.6	The graph classes $\mathcal{S}_5^0, \mathcal{S}_5^1, \mathcal{S}_5^2$ and \mathcal{S}_5^3	147
7.7	The graph $H \in \mathcal{K}_{4,4} - 6e$	150
7.8	The graph $H \in \mathcal{K}_{4,4} - 6e$ with one isolated vertex	151
7.9	Subgraphs $H \in \mathcal{K}_{4,4} - 6e$ with an isolated vertex and no private neighbours	151
7.10	The graph $H \in \mathcal{K}_{4,4} - 6e$ with no isolated vertices and one private neighbour	152

7.11	The graph H with no isolated vertices and no vertices with private neighbours . .	152
7.12	A member of $\mathcal{K}_{r,r} - 2(r-1)e$	153
7.13	Subgraphs of $\mathcal{K}_{3,3,3}$ with $\Delta(H) = 6$	153
7.14	Subgraphs of $\mathcal{K}_{3,3,3}$ with $\Delta(H) = 6$ and $\delta(H) = 1$	154
7.15	Subgraphs of $\mathcal{K}_{3,3,3}$ with $\Delta(H) = 5$ and $\delta(H) = 2$	154
7.16	Subgraphs of $\mathcal{K}_{3,3,3}$ with $\Delta(H) = 5$ and $\delta(H) = 2$	155
7.17	Subgraphs of $\mathcal{K}_{3,3,3}$ with $\Delta(H) = 5$ and $\delta(H) = 2$	155
7.18	A graph $H \subseteq \mathcal{K}_{3,3,3} - 8e$ for which $\gamma_s(H) = 4$	156
7.19	A member of $\mathcal{K}_{r,r,r} - (3r-2)e$	156
7.20	A subgraph of H for which x, y, z are three consecutive vertices on cycle of \overline{H} . .	158
7.21	The graph $\mathcal{K}_{3,3,3} + e$, which admits $\{x, y\}$ as secure dominating set	159
7.22	The labelled book graph $\mathcal{B}_n = \mathcal{K}_2 + \overline{\mathcal{K}_{n-2}}$	160
8.1	The edge-alteration metagraph of the path graph \mathcal{P}_4 of order 4	171

List of Tables

2.1	Truth table for the boolean function $\phi^*(x_1, x_2, x_3, x_4)$ in (2.3)	37
2.2	Worst-case time complexities of algorithms for computing the domination number	43
4.1	Certificates showing that $\gamma_s(G) = 2$ for various infinite classes of graphs G	69
4.2	Certificates showing that $\gamma_s(G) \leq 3$ for various infinite classes of graphs G	72
5.1	Numerical results for grid graphs of orders not exceeding 25	88
5.2	Numerical results for circulant graphs of orders not exceeding 11	89
5.3	Results for grid graphs and hexagonal graphs of orders not exceeding 100	91
5.4	The six linear arrays maintained during execution of Algorithm 5.6	106
5.5	Comparison of execution times of three algorithms when computing γ_s for trees	107
6.1	The cost functions $c_q(\mathcal{P}_6)$ and $C_q(\mathcal{P}_6)$ for the path \mathcal{P}_6	113
7.1	Computation times for constructing edge-removal metagraphs \mathcal{K}_n of small orders	140
7.2	Cardinalities of the graph classes $\mathcal{Q}_n^1, \dots, \mathcal{Q}_n^{\binom{n}{2}}$ for $n \in \{2, \dots, 9\}$	144
7.3	Cardinalities of the graphs classes $\mathcal{S}_n^1, \dots, \mathcal{S}_n^{\binom{n}{2}}$ for $n \in \{2, \dots, 9\}$	148
7.4	The cost function $\mathbf{c}(\mathcal{B}_n)$ and $\mathbf{C}(\mathcal{B}_n)$ for the book graph $\mathcal{B}_n = \mathcal{K}_2 + \overline{\mathcal{K}_{n-2}}$	161

List of Algorithms

2.1	Floyd's Algorithm	19
2.2	MSC	40
2.3	Tree domination	44
5.1	HEURISTIC	80
5.2	BR	81
5.3	FINDBR	82
5.4	BB	85
5.5	FINDBB	85
5.6	DEFENDTREE	104
6.1	An upper bound on the sequence $c(\mathcal{K}_n)$ for a complete graph of order n	133
7.1	Computing the class of \mathcal{Q}_n^q of q -critical graphs of order n	142
7.2	q -CRITICAL($H + e^*, q$)	142
7.3	The class \mathcal{S}_n^p of p -stable graphs of order n for $p > 0$	146
7.4	p -STABLE(G, p)	146

CHAPTER 1

Introduction

Contents

1.1	Dominating and defending chessboards	1
1.2	Casting the net wider: Informal problem description	7
1.3	Dissertation scope and objectives	10
1.4	Dissertation organisation	11

1.1 Dominating and defending chessboards

According to the rules of chess, a queen can (in one move) advance any number of squares horizontally, vertically or diagonally on a chessboard, assuming that no other chess piece lies in its way. For example, the possible squares to which a queen, placed at C6, can move on a standard 8×8 chessboard is shown (in grey shade) in Figure 1.1. These squares are said to be *dominated* by the queen at C6. The following interesting question arises: What is the smallest number of queens that can be placed on an 8×8 chessboard so that every square is dominated by at least one queen?

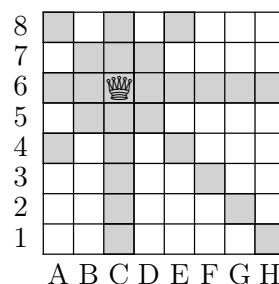


FIGURE 1.1: The queen stationed at C6 can dominate all the squares indicated in grey.

It is easy to see that the answer to this question is at most eight — a bound which may be achieved by placing eight queens in a single row of the board, as shown in Figure 1.2(a). This upper bound may, however, be improved to seven by placing seven queens on all but the last square of any main diagonal of the board, as shown in Figure 1.2(b). By placing six queens as shown in Figure 1.2(c), the upper bound of seven can be improved further to six. All the squares of the board can even be dominated by five queens, placed as shown in Figure 1.2(d), but this process of upper bound improvement cannot be continued beyond five. Indeed, five is

the smallest number of queens that can be placed on an 8×8 chessboard so that every square is dominated by at least one queen. For this reason, this facility location puzzle, which was initially considered during the 1850s by chess enthusiasts in Europe, has become known as the *Five Queens Problem* [60, p. 15].

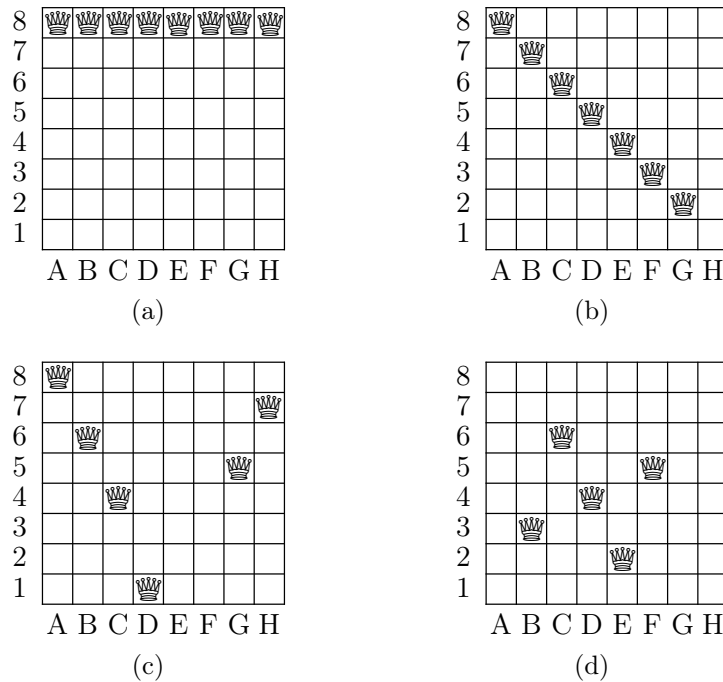


FIGURE 1.2: An example of how (a) eight, (b) seven, (c) six, and (d) five queens may be placed on an 8×8 chessboard to dominate all its squares.

A dynamic protection requirement may be introduced to the Five Queens Problem as follows: What is the smallest number of queens that can be placed on an 8×8 chessboard in such a way that at least one queen can move to any unoccupied square that does not contain a queen, after which every square should again be dominated by the resulting configuration of queens? In such a placement, each square of the board is said to be *defended*. It is clear that the answer to this generalised problem is at least five, since the original placement of the queens (before any move takes place) should, of course, at least dominate all the squares on the board — that is, the requirement that each chessboard square should be defended by at least one queen is stronger than the requirement that each square should be dominated by at least one queen.

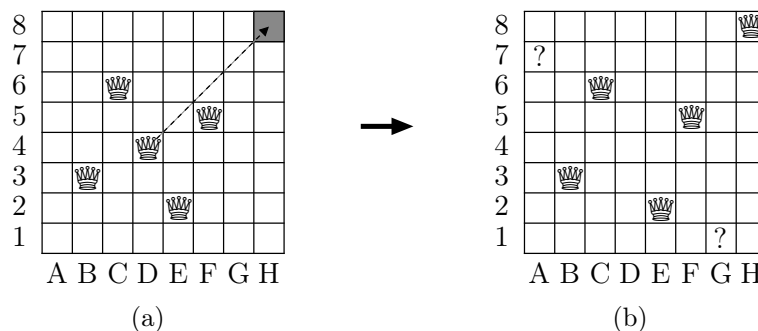


FIGURE 1.3: (a) The queen stationed at D4 is the only queen that can move to H8. (b) If, however, the queen at D4 moves to H8, the squares A7 and G1 are no longer dominated.

Consider again the placement of five queens on an 8×8 chessboard, as indicated in Figure 1.2(d),

and focus on the unoccupied square H8. The queen stationed at D4 is the only queen capable of moving to H8, as shown in Figure 1.3(a). After the movement of the queen from D4 to H8, however, the squares A7 and G1 are no longer dominated — that is, no queen in the placement shown in Figure 1.3(b) can advance to either A7 or G1 in a single move. The placement of queens as indicated in Figure 1.2(d) therefore does not defend all the squares of an 8×8 chessboard. It is, in fact, not possible to find *any* placement of five queens that defends an 8×8 chessboard completely. There is not even a placement of six queens capable of defending an 8×8 chessboard entirely. It is, however, possible to place seven queens on an 8×8 chessboard so as to defend the board completely — such a placement is shown in Figure 1.4.

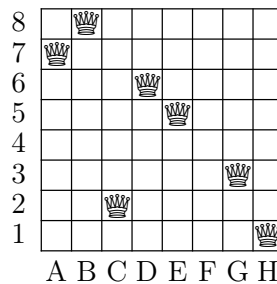


FIGURE 1.4: Seven queens may be placed on an 8×8 chessboard to defend all its squares.

The queens stationed at H1, E5 and B8 are all capable of moving to the unoccupied square H8 in Figure 1.4. However, it is additionally required that all the squares should remain dominated after the movement of a queen to H8. Consider the case where the queen stationed at H1 moves to H8, as shown in Figure 1.5(a). The move leaves the square F1 undominated, as shown in Figure 1.5(b). The queen stationed at H1 therefore does not defend the square H8, although it dominates the square.

Consider next the case where the queen stationed at E5 moves to H8, as shown in Figure 1.5(c). After the move, all the squares remain dominated, as may be verified in Figure 1.5(d). The queen stationed at E5 therefore both dominates and defends the square H8.

Finally, consider the case where the queen stationed at B8 moves to H8, as shown in Figure 1.5(e). After the move, all the squares again remain dominated, as may be verified in Figure 1.5(f). The queen stationed at B8 therefore also both dominates and defends the square H8.

Hence, at least one queen in the placement of Figure 1.4 is capable of defending the unoccupied square H8. This kind of verification may be repeated for the remaining fifty six unoccupied squares in Figure 1.4 so as to verify that the placement of the seven queens in the figure indeed defends the entire board.

The problem of defending the squares of a chessboard using queens may be generalised by also considering other chess pieces, such as kings, bishops, knights and rooks. The movement capabilities of these chess pieces are illustrated in Figure 1.6. The reader may verify that:

- the smallest number of kings required to defend all the squares of an 8×8 chessboard is fourteen, as shown in Figure 1.6(b),
- the smallest number of bishops required to defend all the squares of an 8×8 chessboard is twelve, as shown in Figure 1.6(d),
- the smallest number of knights required to defend all the squares of an 8×8 chessboard is twenty one, as shown in Figure 1.6(f) and

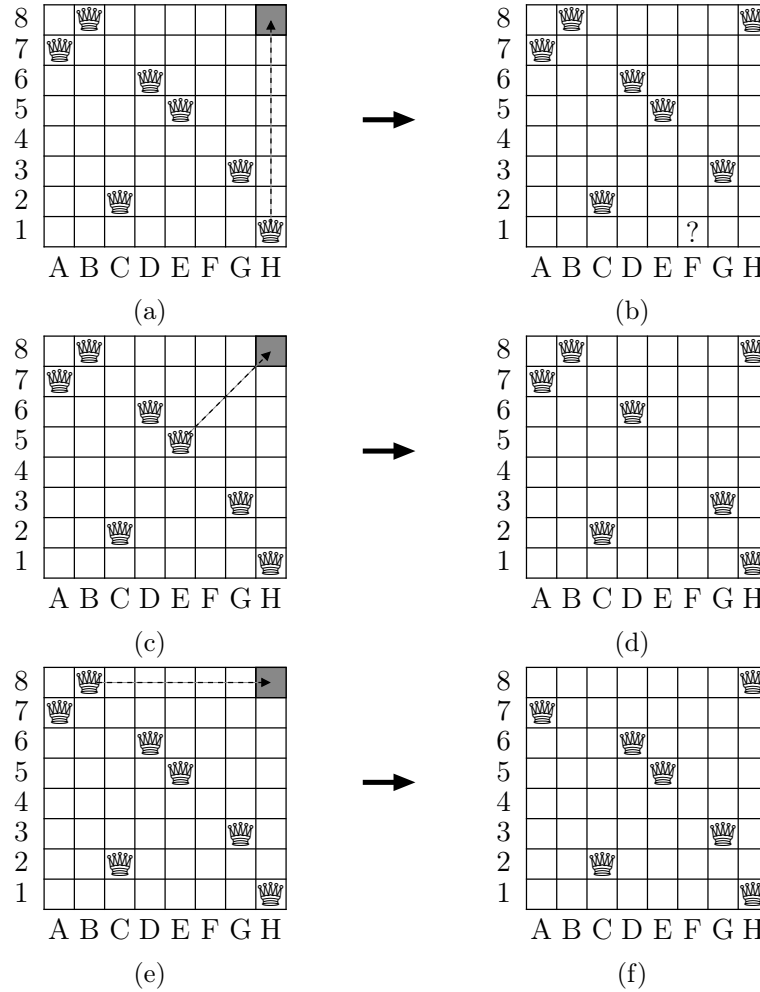


FIGURE 1.5: (a) The queen stationed at H1 can move to the square H8. (b) If, however, the queen at H1 moves to H8, the square F1 is not dominated. (c) The queen stationed at E5 can move to the square H8. (d) If the queen at E5 moves to H8, all the squares of the chessboard remain dominated. (e) The queen stationed at B8 can also move to the square H8. (f) If the queen at B8 moves to H8, all the squares of the chessboard again remain dominated.

- the smallest number of rooks required to defend all the squares of an 8×8 chessboard is eight, as shown in Figure 1.6(h).

Moreover, although the game of chess is traditionally played on an 8×8 board, there is no reason to limit the facility location puzzles described above to boards of this size. The problem of determining the smallest number of queens capable of dominating a chessboard of arbitrary dimensions is called the *Queens Domination Problem*. It is natural to inquire about the rate of increase in the smallest number of queens (or any other chess piece, for that matter) required to defend all the squares of an $n \times n$ board as n grows. This sequence is shown for $n \in \{2, \dots, 9\}$ and all five types of chess pieces in Figure 1.7(a), while instances of corresponding optimal placements are illustrated in Figure 1.7(b)–(i) for the case of placing queens.

Another interesting twist to the facility location puzzles described above occurs when considering obstacles in the line of movement of a chess piece. The example in Figure 1.8(a) shows how a pawn placed at C3 restricts the movement of the rook placed at C6. In this setting, one square fewer has to be defended and many possible moves of the rooks are prohibited by the obstructing pawn. Figure 1.8(b) shows how nine rooks may be placed on the chessboard so as to defend

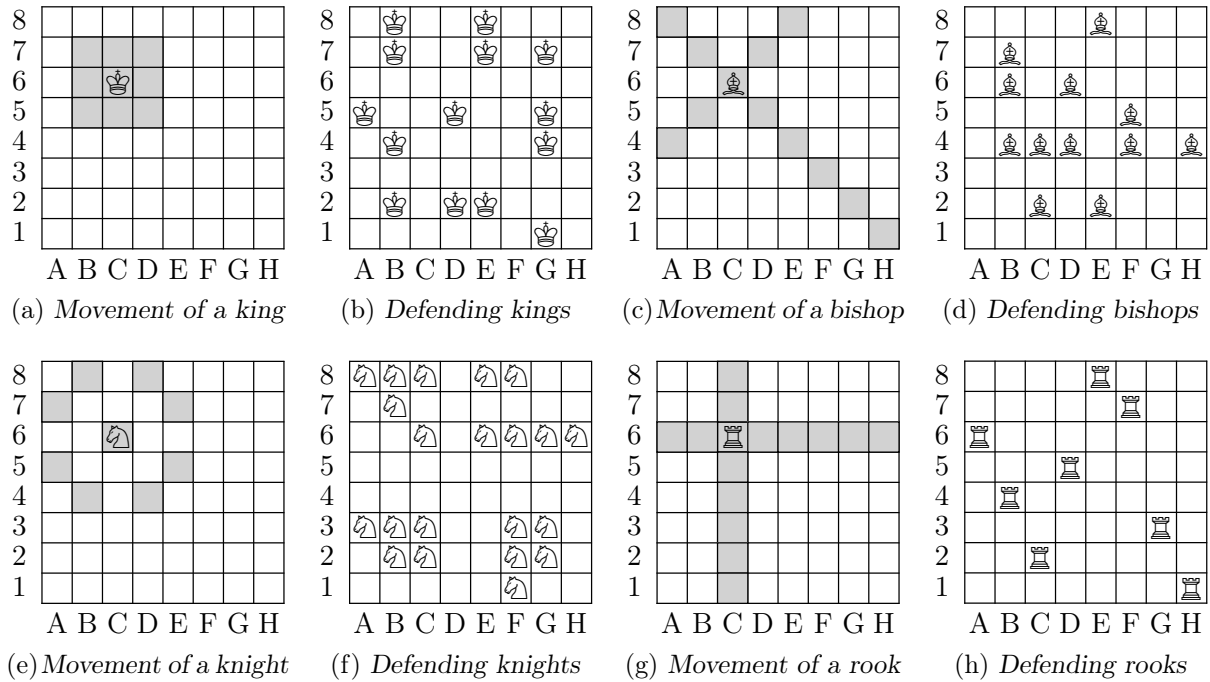


FIGURE 1.6: An example of how (a) fourteen kings, (b) twelve bishops, (c) twenty one knights, and (d) eight rooks may be placed on an 8×8 chessboard to defend all its squares.

n	2	3	4	5	6	7	8	9
Queens	1	2	3	4	5	6	7	8
Kings	1	3	4	6	8	10	14	17
Bishops	2	4	6	7	8	10	12	16
Knights	4	5	6	9	12	16	21	25
Rooks	2	3	4	5	6	7	8	9

(a)

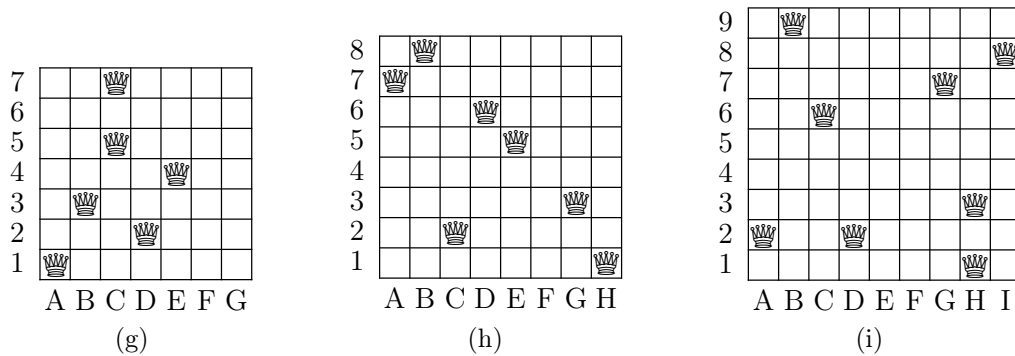
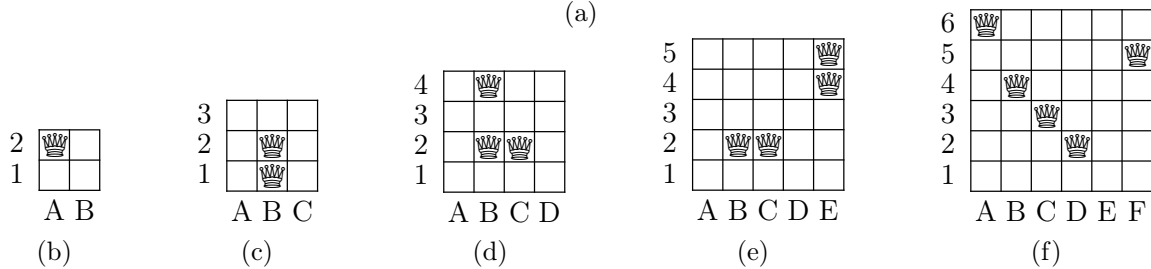


FIGURE 1.7: (a) The smallest number of kings, queens, bishops, knights and rooks required to defend an $n \times n$ chessboard for $n \in \{2, \dots, 9\}$. (b)–(i) Examples of how queens may be placed on an $n \times n$ chessboard so as to defend all the squares for $n \in \{2, \dots, 9\}$.

all the unoccupied squares. As a consequence of the obstructing pawn, an additional rook is required to defend all the unoccupied squares of an 8×8 chessboard. Instead of only considering one obstructing pawn, it is natural to inquire about the rate of increase in the smallest number of rooks required to defend all the unoccupied squares of an 8×8 chessboard as the number of obstructing pawns placed on the board grows. For example, Figure 1.8(c) shows how four pawns placed at C3, D7, E6 and G2 restrict the movement of the rook at C6. In this case four fewer squares need to be defended and a significant set of possible moves by the rooks are prohibited by the obstructing pawns. Figure 1.8(d) shows how ten rooks (an increase of two rooks in addition to the eight of Figure 1.6(h)) may be placed on the chessboard so as to defend all the unoccupied squares of the board.

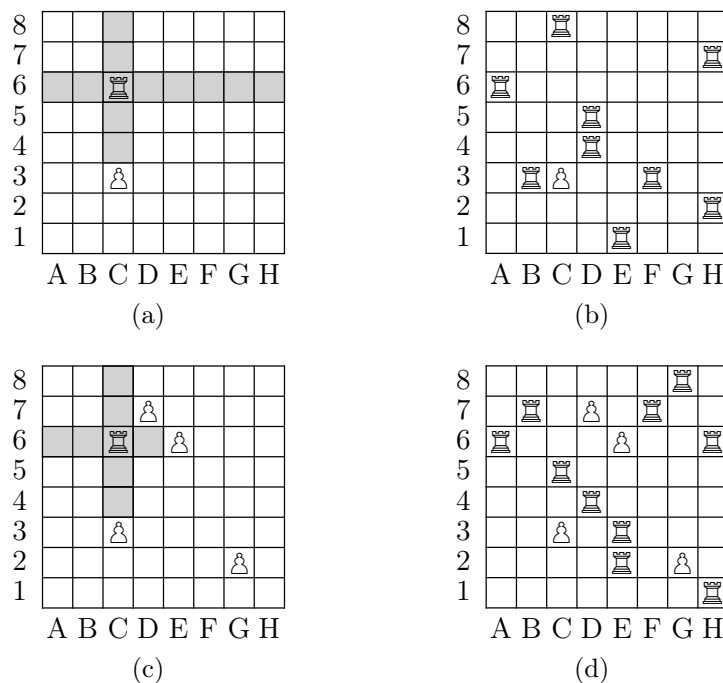


FIGURE 1.8: (a) An example of how the movement of a rook is obstructed by a pawn placed on an 8×8 chessboard. (b) Nine rooks defending all unoccupied squares of a board containing one obstructing pawn. (c) The movement of a rook is obstructed by four pawns. (d) Ten rooks defending all unoccupied squares of a board containing four obstructing pawns.

The reader may have wondered how the author arrived at the various optimal defence placements of the chess pieces presented in the figures of this section. Whereas it is comparatively easy to test the validity (in terms of the defence requirements) of the various placements presented in the figures of this section, it would seem a much harder problem to verify that these placements actually involve the smallest possible number of chess pieces in each case. Indeed, the establishment of an algorithmic procedure for *proving* the non-existence of defence placements involving fewer chess pieces than the claimed minimum cardinalities of the placements presented in this chapter is called for, rather than having to take these claims at face value or having to concur with these claims as a result of one's own inability to find smaller placements upon a series of attempts in a trail-and-error fashion.

1.2 Casting the net wider: Informal problem description

A static variation and a dynamic variation on the notion of chessboard protection by the placement of multiple copies of the same type of chess piece were described in the previous section. There is, however, no reason why protective facility placement puzzles of this nature should be limited to this restrictive context. A more general context for these puzzles is elegantly facilitated within the realm of graph theory.

To see how this generalisation of context may be conceptualised, associate each square of an $n \times n$ chessboard with a vertex of a graph in which two vertices are adjacent if a queen can advance (in one move) from the square corresponding to one of these vertices to the square corresponding to the other vertex. The graph obtained in this manner is called the *queen's graph* (for an $n \times n$ chessboard) and is denoted by Q_n . A king's graph (denoted by K_n), a bishop's graph (denoted by B_n), a knight's graph (denoted by \mathcal{N}_n) and a rook's graph (denoted by \mathcal{R}_n) may be constructed in a similar fashion, by considering the movement capabilities of these alternative chess pieces instead of that of a queen. The graphs Q_4 , K_4 , B_4 , \mathcal{N}_4 and \mathcal{R}_4 (for a 4×4 chessboard) are shown in Figure 1.9.

The facility location puzzles of the previous section may now be recast in the language of graph theory. Let G be a graph with vertex set $V(G)$ and let X be a subset of $V(G)$. A vertex $v \in V(G)$ *dominates* itself as well as all the vertices of G that are adjacent to v . The set X is a *dominating set* of G if each vertex of G is dominated by at least one vertex of X . The minimum cardinality of a dominating set of G is called the *domination number* of G and is denoted by $\gamma(G)$. The smallest number of chess pieces required to dominate all the squares of an 8×8 chessboard is therefore $\gamma(Q_8) = 5$ queens or $\gamma(K_8) = 9$ kings or $\gamma(B_8) = 8$ bishops or $\gamma(\mathcal{N}_8) = 12$ knights or $\gamma(\mathcal{R}_8) = 8$ rooks.

It is widely believed that graph domination originated as a field of study when the Queens Domination Problem was formally proposed by De Jaenisch in 1862 [39]. The problem's significance lies partly in the fact that it was the first "practical" instance of the graph domination problem documented in the literature. A full century would, however, elapse before the notion of domination would be formalised as a graph theoretic concept by Berge [6] and Ore [80] in 1962. Ore [80] was the first to use the terms *dominating set* and *domination number*, and Cockayne and Hedetniemi [33] were the first to use the notation $\gamma(G)$ for the domination number of a graph G .

The dynamic element introduced into the concept of domination to arrive at the generalised notion of defence, as described informally in the previous section, is encapsulated by the graph theoretic notion of *secure domination* due to Cockayne *et al.* [32] which dates back to 2005. Let G be a graph with vertex set $V(G)$ and let X be a subset of $V(G)$. Then X is a *secure dominating set* of G if, for each vertex u in $V(G) - X$, there exists a neighbouring vertex v of u in X such that $(X - \{v\}) \cup \{u\}$ is a dominating set of G (in which case v is said to *defend* u). The minimum cardinality of a secure dominating set of G is called the *secure domination number* of G , and is denoted by $\gamma_s(G)$.

The smallest number of chess pieces required to defend all the squares of an 8×8 chessboard is therefore $\gamma_s(Q_8) = 7$ queens or $\gamma_s(K_8) = 14$ kings or $\gamma_s(B_8) = 12$ bishops or $\gamma_s(\mathcal{N}_8) = 21$ knights or $\gamma_s(\mathcal{R}_8) = 8$ rooks.

Applications of the notion of secure domination abound: If the vertices of the graph G denote geographically dispersed facilities, and the edges model links between these facilities along which patrolling guards may move, then a secure dominating set of G represents a collection of facility locations at which guards may be placed so that the entire complex of facilities modelled by G is protected (in the sense that if a security problem were to occur at facility u , there will either be

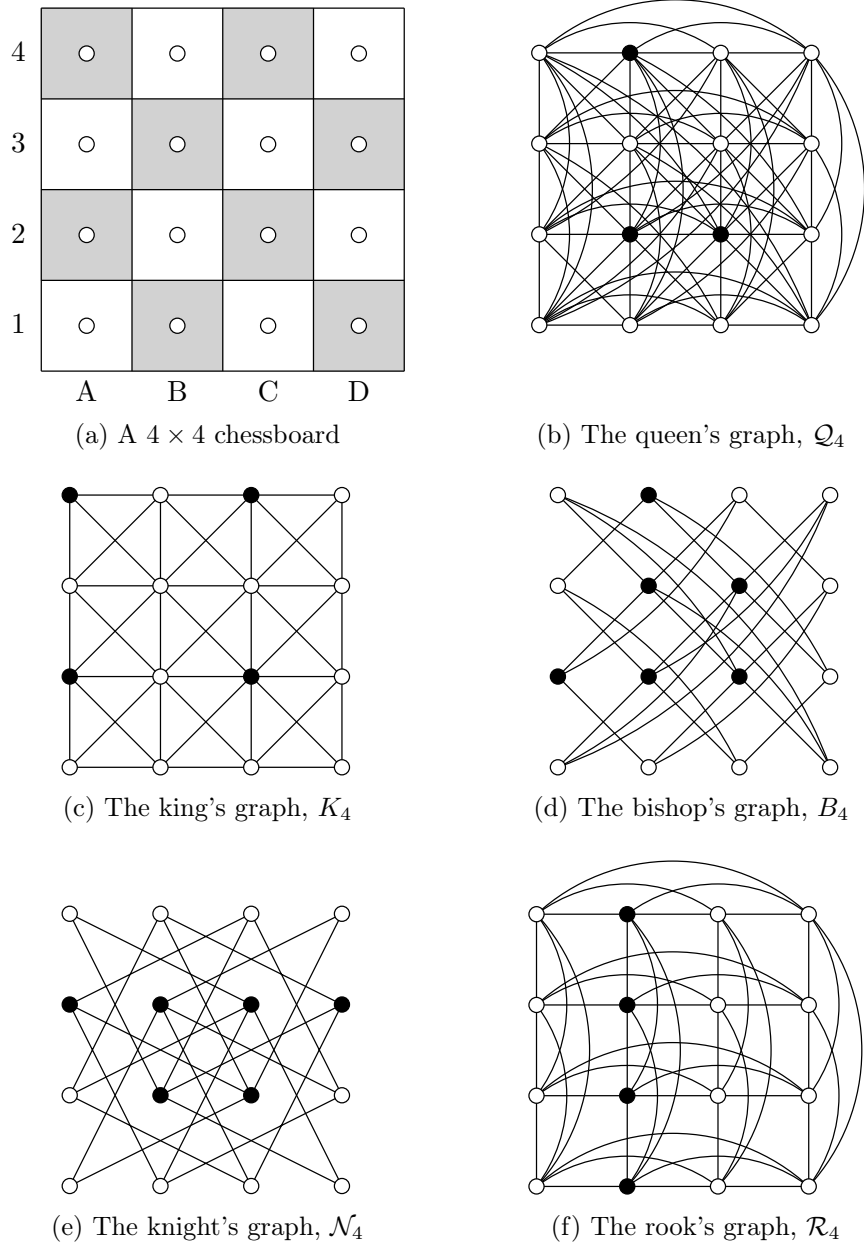


FIGURE 1.9: (a) A 4×4 chessboard. (b) The corresponding queen's graph, \mathcal{Q}_4 . (c) The corresponding king's graph, K_4 . (d) The corresponding bishop's graph, B_4 . (e) The corresponding knight's graph, \mathcal{N}_4 . (f) The corresponding rook's graph, \mathcal{R}_4 . Minimum secure dominating sets for \mathcal{Q}_4 , K_4 , B_4 , \mathcal{N}_4 and \mathcal{R}_4 are indicated by the solid vertices in (b)–(f).

a guard at that facility who can deal with the problem, or else a guard dealing with the problem from an adjacent facility v will leave the facility complex dominated after moving from facility v to facility u in order to deal with the problem). In this application, the secure domination number represents the minimum number of guards required to protect the facility complex.

The generic application above may prompt one to wonder what kind of edge structure in the graph model makes it intrinsically cheap or expensive (in terms of the smallest number of guards required) to dominate the graph securely, or, more formally:

Question 1.1 *What can be said about the structure of graphs which admit small secure dominating sets? For example, is it possible to characterise graphs with secure domination number two?*

As alluded to in §1.1, verification of whether a given subset of the vertex set of a graph G is a secure dominating set of G is relatively easy (this can be achieved in polynomial time in terms of the number of vertices of G). Finding the minimum cardinality of a secure dominating set of G is, however, a much harder problem. This raises the following question:

Question 1.2 *What is the most efficient way of computing the secure domination number of an arbitrary graph?*

Prompted by the example of obstructing pawns in Figure 1.8, one might also seek the cost (in terms of the additional number of guards required over and above the minimum number, $\gamma_s(G)$, to protect an entire location complex modelled by a graph G in the secure dominating sense) if a number of edges and/or vertices of G were to “fail” (*i.e.* a number of links and/or placement locations were to be eliminated from the graph modelling accessibility between the locations in the complex, so that guards may no longer move along such disabled links and/or may no longer be stationed at such disabled vertices).

Two cost functions, denoted by $c_q(G)$ and $C_q(G)$, may, for instance, be employed to measure respectively the *smallest* possible and the *largest* possible increase in the minimum number of guards required to dominate a graph G with m edges securely upon the removal of $q \in \{1, \dots, m\}$ arbitrary edges. For example, the cost functions $c_q(\mathcal{R}_3)$ and $C_q(\mathcal{R}_3)$ for the rook’s graph \mathcal{R}_3 (which contains nine vertices and eighteen edges) are shown in Figure 1.10. The following question arises naturally:

Question 1.3 *What can be said about the shapes and growth rates of the functions $c_q(G)$ and $C_q(G)$ as q increases for an arbitrary graph G ?*

A graph G is *q-critical* if the *smallest* arbitrary subset of edges whose removal from G necessarily increases the secure domination number of the resulting graph, has cardinality q . In terms of the cost function c described above, a graph G is therefore *q-critical* if $c_1(G) = c_2(G) = \dots = c_{q-1}(G) = 0$, but $c_q(G) > 0$. The rook’s graph \mathcal{R}_3 , for example, is 9-critical. Being able to determine the value of q for which a given graph is *q-critical* is important from an application point of view, because this value may be seen as a robustness threshold in the sense that the failure of some set of $q - 1$ edges in G results in a graph that can still be dominated securely by $\gamma_s(G)$ guards, but this is not true for the failure of *any* set of q edges in G . The notion of criticality raises the following interesting question:

Question 1.4 *How would one go about constructing the class of *q-critical* graphs of order n ?*

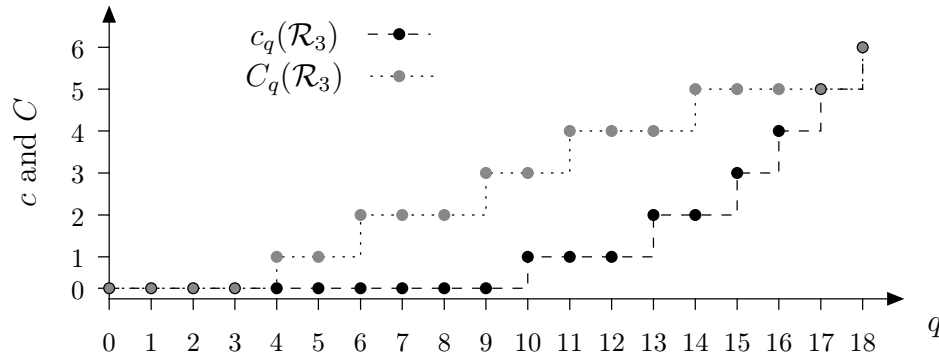


FIGURE 1.10: The cost functions $c_q(\mathcal{R}_3)$ and $C_q(\mathcal{R}_3)$ for $q \in \{0, \dots, 18\}$.

In a similar vein, a graph G is p -stable if the *largest* subset of arbitrary edges whose removal from G necessarily does *not* increase the secure domination number of the resulting graph, has cardinality p . In terms of the cost function C described above, a graph is p -stable if $C_0(G) = C_1(G) = \dots = C_p(G) = 0$, but $C_{p+1}(G) > 0$. The rook's graph \mathcal{R}_3 , for example, is 3-stable. The concepts of stability and criticality are, in a sense, therefore dual notions, and being able to determine the value of p for which a given graph G is p -stable is important from an application point of view, because this value may be seen as a robustness threshold in the sense that the failure of any $p + 1$ edges in G results in a graph that cannot be dominated securely by $\gamma_s(G)$ guards, but there is some set of p edges for which this is not true. The notion of stability raises the following interesting question:

Question 1.5 *How would one go about constructing the class of p -stable graphs of order n ?*

If a graph with m edges is both p -stable and q -critical, then clearly $0 \leq p \leq q \leq m$. The following question, however, seems interesting:

Question 1.6 *Can the difference $q - p$ between the values q and p for which a graph can be q -critical and p -stable be bounded from above? Or can this difference be arbitrarily large?*

Finally, the upper bound of m on the values of p and q for which a graph with m edges can be p -stable and q -critical seems overly excessive. For example, the rook's graph \mathcal{R}_3 has eighteen edges, yet it is only 3-stable and 9-critical. This raises the following question:

Question 1.7 *What is the largest value of p for which a graph with n vertices and m edges can be p -stable? Similarly, what is the largest value of q for which a graph with n vertices and m edges can be q -critical?*

Answers to questions such as Questions 1.1–1.7 are pursued in this dissertation.

1.3 Dissertation scope and objectives

More generally, the following seven objectives are pursued in this dissertation:

- I To *document* and *interpret* the literature related to the notion of secure graph domination.

- II To *characterise* those graphs with small secure domination numbers.
- III To *design* and *implement* algorithms for computing the exact values of the secure domination number of a graph and to *analyse* the time and space complexities of these algorithms.
- IV To *establish* good general bounds on the cost functions c_q and C_q described above for arbitrary graphs.
- V To *propose* tighter bounds on or exact values of the cost functions c_q and C_q for various infinite graph classes of special structure, such as paths, cycles, wheels, complete graphs and complete bipartite graphs.
- VI To *develop* and *implement* techniques for computing the classes of p -stable and q -critical graphs of order n for small values of n and all admissible values of p and q .
- VII To *pose* a number of open questions and intriguing problems related to the notions of p -stability and q -criticality in secure graph domination.

Graphs other than simple, undirected graphs are considered to fall beyond the scope of this dissertation. Furthermore, only the notion of secure domination is considered (although the context for this field of study is established by referring briefly in the dissertation to other protection parameters). Finally, the notions of p -stability and q -criticality will only be considered in the context of edge removal (although it would also be possible to carry out similar studies in the contexts of edge insertion and vertex removal).

1.4 Dissertation organisation

The second chapter of this dissertation lays down the fundamental groundwork from the realms of graph theory and complexity theory required to understand the general exposition of later chapters. A number of basic definitions from graph theory are provided in the first section, while fundamental concepts from graph domination are introduced in the second section. The notions of domination, independence and irredundance are described, and this is followed by a review of a series of well-known results on the domination number of a graph. Furthermore, results in the literature on the effects of edge removal are investigated in the context of graph domination. Basic notions from complexity theory are reviewed in the third section. The complexity classes **P** and **NP** are described, and a proof of the well-known result that the dominating set decision problem is **NP**-complete is recounted. Some results related to determining the domination number of a graph are also presented.

In the third chapter, a literature review of known results on secure graph domination is presented. In the first section, a number of graph protection parameters related to the secure domination number of a graph are described, including those associated with classical domination, total domination, Roman domination and weak Roman domination. The section closes with an inequality chain relating these parameters, which is due to Cockayne *et al.* [32]. In the second section, various known results from the literature on secure graph domination are presented, including a number of general bounds on the secure domination number, and exact values of the secure domination number for certain infinite classes of graphs. The notion of edge removal in secure graph domination is finally considered, specifically with a focus on stability and criticality in secure graph domination. The chapter closes with a description of a number of variations on the notion of secure graph domination.

The main focus of Chapter 4 is to contribute a number of basic results on the nature and computation of minimum secure dominating sets of arbitrary graphs. The chapter opens in §4.1 with a description of three necessary and sufficient criteria for establishing whether or not a given subset of the vertex set of a graph is, in fact, a secure dominating set of the graph. Using these criteria, the classes of graphs that have secure domination number 1, 2 or 3 are then characterised in §4.2. A result is presented in §4.3 which states that it is possible to successively increase the number of defenders in any minimum secure dominating set of a connected graph with minimum degree at least two, until all the members of the set are defenders. Finally, the decision problem associated with the problem of computing the secure domination number of an arbitrary graph is shown to be **NP-complete** in §4.4.

Chapter 5 opens with a presentation of three algorithmic approaches towards computing the secure domination number of an arbitrary graph, including a branch-and-bound algorithm, a branch-and-reduce algorithm and a binary programming problem formulation. A linear algorithm is finally presented for computing the secure domination number of an arbitrary tree. The chapter closes with a brief appraisal of the relative performances of the four algorithmic approaches.

The focus of Chapter 6 falls on determining the smallest and largest increase in the secure domination number of a graph when edges are removed from the graph. The chapter opens in §6.1 with basic results on the cost functions c_q and C_q introduced in §1.2. Known bounds on the secure domination number of a graph are reviewed and some new bounds are established in §6.2. These bounds are then used to derive general bounds on the two cost functions c_q and C_q . The remainder of the chapter is devoted to determining exact values of or tight bounds on the cost functions c_q and C_q for various infinite graph classes.

In Chapter 7, threshold information is presented on the number of edge removals from a graph before increasing its secure domination number. The chapter opens in §7.1 with formal descriptions of the notions of criticality and stability, introduced informally in §1.2. A characterisation of q -critical graphs is established in §7.2, which is used to compute all q -critical graphs of small order inductively. Similar results are established in §7.3 for the notion of stability. An investigation into the largest possible values of p and q for which a graph of order n can be p -stable and q -critical is conducted in §7.4, while the exact values of p and q for which members of various infinite classes of graphs of special structure are p -stable and q -critical are presented in §7.5.

The dissertation closes in §8.1 with a summary of the work presented within, an appraisal of the contributions of the dissertation in §8.2, and a number of ideas with respect to related future work in §8.3.

CHAPTER 2

Mathematical preliminaries

Contents

2.1	Basic notions in graph theory	13
2.1.1	Neighbourhoods	14
2.1.2	Graph isomorphisms and subgraphs	15
2.1.3	Connected Graphs	16
2.1.4	Graph operations and special graphs	20
2.2	Fundamentals of graph domination	24
2.2.1	Domination, independence and irredundance	24
2.2.2	The domination number of a graph	26
2.2.3	Multiple domination	30
2.2.4	Edge criticality in graph domination	30
2.3	Basic notions of complexity theory	33
2.3.1	The complexity classes P and NP	34
2.3.2	Reductions and NP -completeness	35
2.3.3	The satisfiability decision problem	36
2.3.4	From decision problems to computation problems	37
2.3.5	Graph domination is NP -complete	38
2.3.6	Algorithms for computing the domination number of a graph	39
2.4	Chapter summary	44

This chapter provides the necessary mathematical background in order to facilitate understanding of the concepts and ideas employed throughout the remainder of the dissertation. Basic concepts in graph theory are reviewed in §2.1. This is followed by a more focussed overview of certain fundamental concepts in the theory of graph domination in §2.2. An overview of certain basic notions in complexity theory are finally presented in §2.3.

2.1 Basic notions in graph theory

A *graph* $G = (V, E)$ is a nonempty, finite set $V(G)$ of elements, called *vertices* (the singular being *vertex*), together with a (possibly empty) set $E(G)$ of two-element subsets of $V(G)$, called *edges* of G . The number of vertices in a graph G is called the *order* of G , denoted by $n = |V(G)|$, while the number of edges in a graph G is called the *size* of G , denoted by $m = |E(G)|$.

A graphical representation of a graph G_1 with vertex set $V(G_1) = \{v_1, \dots, v_7\}$ and edge set $E(G_1) = \{v_1v_3, v_1v_5, v_1v_7, v_2v_3, v_2v_5, v_2v_7, v_3v_4, v_3v_5, v_3v_7, v_4v_7\}$ is shown in Figure 2.1(a).

The edge $e = \{u, v\}$ joins the vertices u and v . If $e = \{u, v\}$ (informally written as $e = uv$) is an edge of G , then u and v are *adjacent* vertices said to be *joined* by e , while u and e are *incident*, as are v and e . Two vertices that are not joined by an edge are *nonadjacent*. In the graph G_1 in Figure 2.1(a) the vertices v_1 and v_5 are adjacent, while the edge v_1v_3 is incident with the vertex v_1 . The edges v_2v_3 and v_2v_5 are both incident with the vertex v_2 .

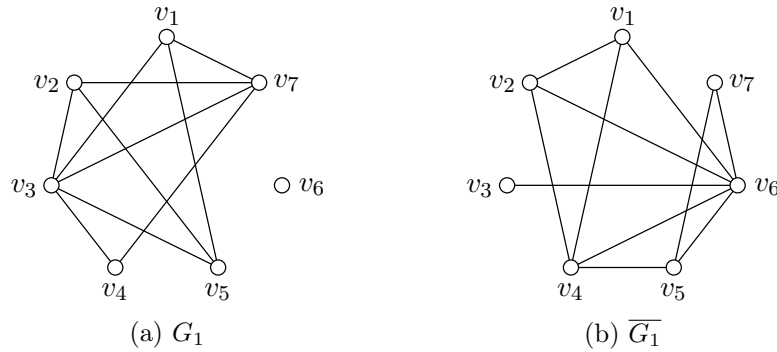


FIGURE 2.1: (a) Graphical representation of a graph G_1 with vertex set $V(G_1) = \{v_1, \dots, v_7\}$ and edge set $E(G_1) = \{v_1v_3, v_1v_5, v_1v_7, v_2v_3, v_2v_5, v_2v_7, v_3v_4, v_3v_5, v_3v_7, v_4v_7\}$. (b) The complement $\overline{G_1}$ of the graph G_1 .

The complement \overline{G} of a graph $G(V, E)$ is a graph with vertex set $V(\overline{G}) = V(G)$ and for which $e \in E(\overline{G})$ if and only if $e \notin E(G)$. The complement $\overline{G_1}$ of the graph G_1 in Figure 2.1(a), shown in Figure 2.1(b), has vertex set $V(\overline{G_1}) = V(G_1)$ and edge set $\{v_1v_2, v_1v_4, v_1v_6, v_2v_4, v_2v_6, v_3v_6, v_4v_5, v_4v_6, v_5v_6, v_5v_7, v_6v_7\}$.

2.1.1 Neighbourhoods

The *open neighbourhood* of a vertex v of the graph G , denoted $N_G(v)$ (or $N(v)$ if G is clear from the context), is the set

$$N_G(v) = \{u \in V(G) \mid uv \in E(G)\}.$$

The *closed neighbourhood*, of a vertex v of G , denoted by $N_G[v]$ or $N[v]$, is the set

$$N_G[v] = N_G(v) \cup \{v\}.$$

For any vertex v of a graph G , the number of vertices adjacent to v , i.e. $|N_G(v)|$, is called the *degree* of v in G and is denoted by $\deg_G(v)$. A vertex v with $\deg_G(v) = 0$ is called an *isolated vertex* of G , while a vertex with $\deg_G(v) = 1$ is called an *end-vertex* of G . A *universal vertex* of a graph G is a vertex v that is adjacent to all vertices in $V(G) - \{v\}$. A vertex adjacent to an end-vertex is called a *support vertex*. A vertex is called *odd* or *even* depending on whether its degree is odd or even. The *minimum degree* of a graph G is the minimum degree among all the vertices of G and is denoted by $\delta(G)$, while the *maximum degree* of G is the maximum degree among all the vertices of G and is denoted by $\Delta(G)$, that is $\delta(G) = \min_{i \in \{1, \dots, n\}} \{\deg_G(v_i)\}$ and $\Delta(G) = \max_{i \in \{1, \dots, n\}} \{\deg_G(v_i)\}$, where $V(G) = \{v_1, \dots, v_n\}$.

The open neighbourhood of the vertex v_2 in the graph G_1 in Figure 2.1(a) is $N_{G_1}(v_2) = \{v_3, v_5, v_7\}$, while its closed neighbourhood is $N_{G_1}[v_2] = \{v_2, v_3, v_5, v_7\}$. The vertex v_6 is an

isolated vertex, and there is no end-vertex in G_1 . The vertex v_6 in Figure 2.1(b) is a universal vertex of $\overline{G_1}$. Furthermore, $\delta(G_1) = 0$ and $\Delta(G_1) = 5$.

Observe that the sum of the degrees of the seven vertices of the graph G_1 in Figure 2.1(a) is 20. The following result, often referred to as the fundamental theorem of graph theory, relates the sum total of the degrees and the size of any graph [25, p. 6].

Theorem 2.1 (Fundamental Theorem of Graph Theory) *Let G be a graph of order n and size m , and let $V(G) = \{v_1, \dots, v_n\}$. Then*

$$\sum_{i=1}^n \deg_G(v_i) = 2m.$$

■

An important consequence of Theorem 2.1 is given in the next result for which a proof may be found in [63, pp. 4–5].

Corollary 2.1 *In any graph there is an even number of odd vertices.*

■

An even number of vertices in the graph G_1 in Figure 2.1(a) have odd degree (the four vertices v_1, v_2, v_3 and v_5), while the number of even vertices in a graph can be odd or even (the three vertices v_4, v_6 and v_7 are even).

2.1.2 Graph isomorphisms and subgraphs

Two graphs G and H are *isomorphic*, denoted by $G \cong H$, if there exists a one-to-one mapping $\phi : V(G) \mapsto V(H)$ such that $uv \in E(G)$ if and only if $\phi(u)\phi(v) \in E(H)$. In this case the function ϕ is called an *isomorphism*. If ϕ maps the graph G onto itself, ϕ is called an *automorphism*. Two graphs G and H are *equal* if $V(G) = V(H)$ and $E(G) = E(H)$. Equal graphs are therefore isomorphic, but the converse is not necessarily true. An illustration of the notions of an isomorphism between and of equality of graphs may be found in Figure 2.2.

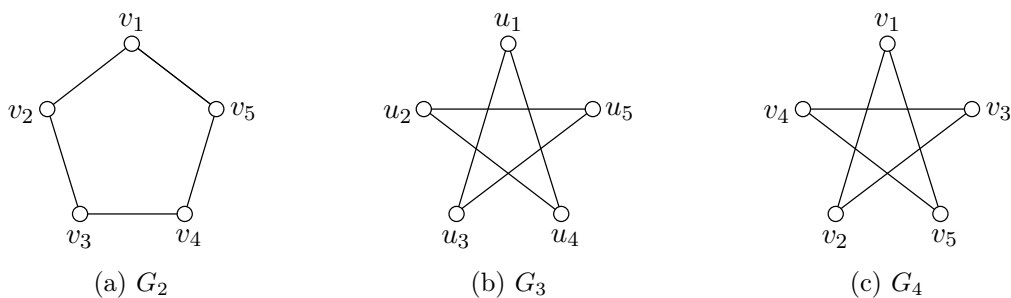


FIGURE 2.2: The graph G_3 in (b) is isomorphic (but not equal) to G_2 in (a), an isomorphism $\phi : V(G_2) \mapsto V(G_3)$ being $\phi(v_1) = u_1$, $\phi(v_2) = u_3$, $\phi(v_3) = u_5$, $\phi(v_4) = u_2$ and $\phi(v_5) = u_4$. The graph G_4 is both equal and isomorphic to G_2 .

A graph H is a *subgraph* of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, and a *spanning subgraph* of G if $V(H) = V(G)$ and $E(H) \subseteq E(G)$. Any graph G is clearly a subgraph of itself and a graph H is called a *proper subgraph* of a graph G , denoted $H \subset G$, if it is a subgraph of G and $H \not\cong G$. For a nonempty vertex subset $S \subseteq V(G)$ of a graph G the *induced subgraph* of S in G , denoted by $\langle S \rangle_G$, is the subgraph of G with vertex set S and the edge set

$E(\langle S \rangle_G) = \{uv \in E(G) \mid u, v \in S\}$. The notions of a subgraph, a spanning subgraph and an induced subgraph are illustrated in Figure 2.3.

A graph F , a graph G is called F -free if G does not contain an induced subgraph isomorphic to F . If $F \cong \mathcal{K}_{1,3}$, an F -free graph is often called *claw-free*.

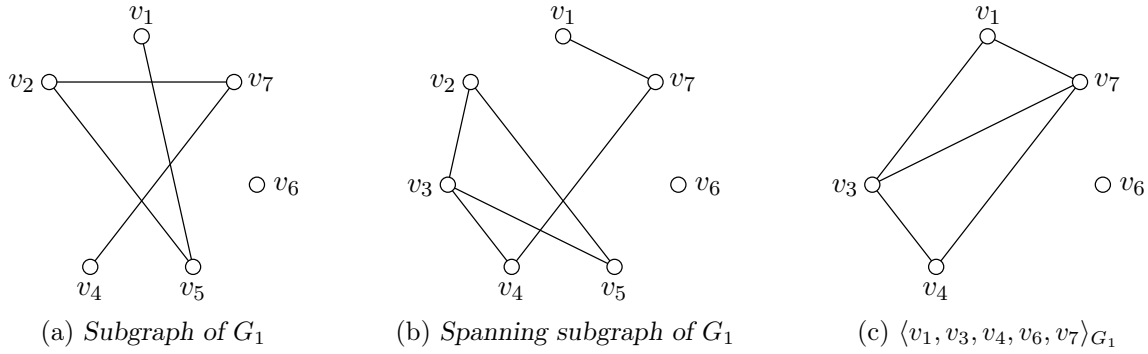


FIGURE 2.3: The graph in (a) is an example of a subgraph of G_1 in Figure 2.1(a), while the graph in (b) is a spanning subgraph of G_1 . The induced subgraph $\langle \{v_1, v_3, v_4, v_6, v_7\} \rangle_{G_1}$ is shown in (c).

2.1.3 Connected Graphs

A v_1 - v_ℓ walk in a graph G is a finite alternating sequence of vertices and edges

$$v_0, e_1, v_1, e_2, v_2, \dots, v_{i-1}, e_i, v_i, \dots, v_{\ell-1}, e_\ell, v_\ell$$

that begins with the vertex v_1 and ends with the vertex v_ℓ ($\ell \geq 0$), such that $e_i = v_i v_{i+1}$ is an edge of G for all $i = 1, \dots, \ell - 1$. For the sake of brevity the edges are often omitted from the sequence when denoting a walk. The number of edges in the walk is called the *length* of the walk. An example of a walk of length 6 in the graph G_1 in Figure 2.1(a) is $v_1, v_3, v_2, v_7, v_3, v_4, v_7$. A v_1 - v_ℓ path in a graph G is a v_1 - v_ℓ walk in G in which no vertex is repeated. The *distance* $d_G(u, v)$ between two vertices u and v in a graph G , is the length of a shortest u - v path in G if such a path exists; otherwise $d_G(u, v) = \infty$. The length of $\max_{u, v \in V(G)} d_G(u, v)$ of a graph G is called the *diameter* of G and is denoted by $\text{diam}(G)$.

If $v_1 = v_n$ in a walk and no other vertices are repeated, the walk is called a *cycle*. A cycle of odd length [even length] is called an *odd cycle* [*even cycle*]. The length of a shortest cycle in a graph G is called the *girth* of G (if such a cycle exists) and is denoted by $g(G)$. If G has no cycles, then $g(G) = \infty$.

In the graph G_1 in Figure 2.1(a), $v_1, v_5, v_2, v_3, v_7, v_4$ is an example of a path of length 5, while v_1, v_3, v_2, v_7, v_1 is an example of a cycle of length 4. Furthermore, $g(G_1) = 3$, with an example of shortest cycle in G_1 being v_1, v_3, v_7, v_1 .

If there exists a u - v path for every pair of vertices u and v of a graph, the graph is said to be *connected*; otherwise it is *disconnected*. A *component* of a graph G is a subgraph of G that is connected and is not a subgraph of any larger connected subgraph of G . The number of components of a graph G is denoted by $k(G)$. A graph G is therefore connected if and only if $k(G) = 1$.

The graph G_1 in Figure 2.1(a) is disconnected, because there is no path between, for example, the vertices v_1 and v_6 . Furthermore, $k(G_1) = 2$ and the graphs $\langle \{v_1, v_2, v_3, v_4, v_5, v_7\} \rangle_{G_1}$ and $\langle \{v_6\} \rangle_{G_1}$ are the two components of G_1 .

A *tree* is a connected graph that contains no cycles, while a *forest* is a graph that has no cycles. It therefore follows that each component of a forest is a tree. A *leaf* of a tree T is an end-vertex of T . Every vertex of a tree T that is not a leaf of T is called an *internal vertex* of T . Any vertex adjacent to at least one leaf of a tree T is called a *support vertex* of T , while an r -support vertex of T is a vertex adjacent to at least r leaves of T . A *spanning tree* of a connected graph G is a tree that is a subgraph of G and contains all the vertices of G . A *spanning forest* of a (possibly disconnected) graph G is a subgraph of G in which each component is a tree and which contains all the vertices of G .

The notions of a spanning tree and spanning forest are illustrated in Figure 2.4. The graph in Figure 2.4(a) contains four leaves v_1, v_3, v_4, v_7 , two internal (and support) vertices v_2, v_5, v_6 , with v_2 and v_5 being a 1-support vertices, while v_6 is a 2-support vertex.

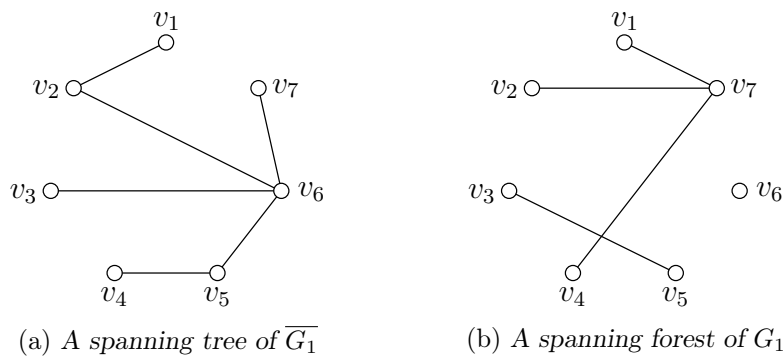


FIGURE 2.4: The graph in (a) is an example of a spanning tree of the graph \overline{G}_1 in Figure 2.1(b), while the graph in (b) is a spanning forest of the graph G_1 in Figure 2.1(a).

A *rooted tree* is a tree in which a single vertex, called the *root* has been specified. The norm when representing rooted trees graphically is to place the root at the top as the only vertex on a horizontal level called *level 0* of the tree. The vertices adjacent to the root are then placed below that level on a next horizontal level called *level 1*, and so forth. Any vertex on level i is therefore adjacent to exactly one vertex on level $i - 1$ and is a distance i from the root. The number of the highest numbered level thus formed is called the *height* of the rooted tree. If v is a vertex on level i of a tree T , then v is the *parent* of all the vertices in $N_T(v)$ on level $i + 1$. Similarly, if a vertex v on level i is adjacent to a vertex u on level $i - 1$, then v is called a *child* of u . A vertex $u \in V(T)$ is a *decendant* [*ancestor*, resp.] of v if there exists a u - v path in T and the level of u is smaller [larger, resp.] than that of v . The notions of the root, the leaves, the internal vertices, the children, the levels and the height of a rooted tree are illustrated in Figure 2.5.

The *deletion* of a nonempty vertex subset $S \subseteq V(G)$ from a graph G is the subgraph with vertex set $V(G) - S$ and edge set $\{uv \in E(G) \mid u, v \notin S\}$. Such a subgraph is denoted by $G - S$. Similarly, for any edge subset $D \subseteq E(G)$ the *deletion* of the edge set D is the spanning subgraph of G with edge set $E(G) - D$ and is denoted by $G - D$. If $S = \{v\}$ [$D = \{e\}$, resp.] for some $v \in V(G)$ [$e \in E(G)$, resp.], the graph $G - S$ [$G - D$, resp.] is simply denoted by $G - v$ [$G - e$, resp.]. The *addition* of a nonempty edge set $D \subseteq E(\overline{G})$ to a graph G is the graph with vertex set $V(G)$ and edge set $E(G) \cup D$. Such a graph is denoted by $G + D$ and if $D = \{e\}$ for some $e \in E(\overline{G})$, then the graph $G + D$ is simply denoted by $G + e$. Considering the graph G_5 in Figure 2.6(a), with vertex subset $S = \{v_2\}$ and edge subset $D = \{v_1v_3, v_2v_3, v_2v_4\}$, the subgraph $G_5 - S$ is shown in Figure 2.6(b), while $G_5 - D$ is shown in Figure 2.6(c).

A graph may be represented by means of an adjacency matrix or weight matrix. The *adjacency matrix* of a graph G of order n , denoted by $A(G)$, is an $n \times n$ binary matrix whose (i, j) -th

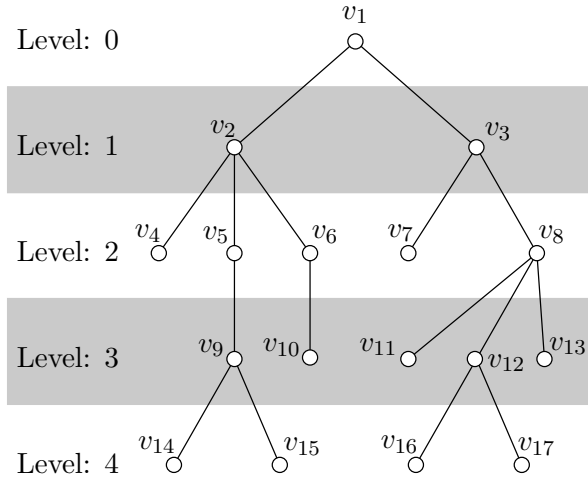


FIGURE 2.5: The vertices $v_1, v_2, v_3, v_5, v_6, v_8, v_9, v_{12}$ are internal vertices of the rooted tree in the figure, while the vertices $v_4, v_7, v_{10}, v_{11}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}$ are the leaves of the tree. The vertices v_{11}, v_{12} and v_{13} are children of the vertex v_8 and the vertex v_1 is both the root of the tree and the parent of vertices v_2 and v_3 . The height of the tree is 4.

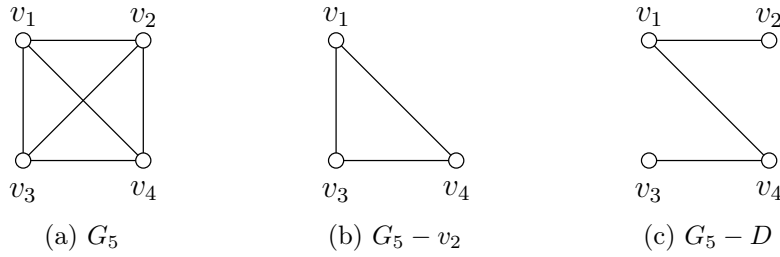


FIGURE 2.6: Illustration of the deletion of (b) a vertex and (c) an edge subset $D = \{v_1v_3, v_2v_3, v_2v_4\}$.

element is 1 if $v_i v_j \in E(G)$ and 0 if $v_i v_j \notin E(G)$. The *weight matrix* of a graph differs from its adjacency matrix in one aspect only, namely that the off-diagonal zero entries of the latter are replaced by entries of infinite magnitude in the former. The adjacency matrix and the weight matrix of a graph are therefore symmetric. The adjacency matrix and the weight matrix of the graph G_6 in Figure 2.7(a) are shown in Figure 2.7(b) and (c), respectively.

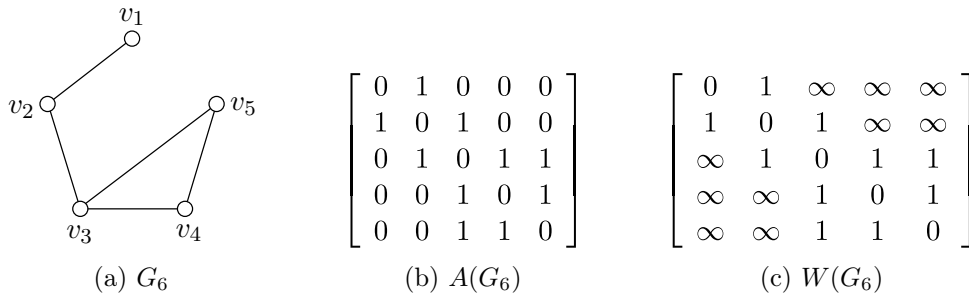


FIGURE 2.7: A graph G_6 , its adjacency matrix $A(G_6)$ and its weight matrix $W(G_6)$.

Floyd's algorithm may be used to find the distances between all pairs of vertices in a graph. A pseudo-code description of this algorithm is given in Algorithm 2.1. Consider a graph G of order n with vertex set $V(G) = \{v_1, \dots, v_n\}$ and suppose that the matrix $D^{(k-1)} = [d_{ij}^{(k-1)}]$ contains as its (i, j) -th entry $d_{ij}^{(k-1)}$ the length of a shortest v_i - v_j path utilising as its internal vertices

vertices from the set $\{v_1, \dots, v_{k-1}\}$ only. Then it is clear that $D^{(0)} = [d_{ij}^{(0)}]$ is the weight matrix of G , while $D^{(n)}$ contains as its (i, j) -th entry the distance between v_i and v_j in G . Furthermore, it is clear that $d_{ij}^{(k)} \leq d_{ij}^{(k-1)}$ for all $i, j, k \in \{1, \dots, n\}$, as an additional vertex (the vertex v_k) is available to include as internal vertex in the v_i - v_j path lengths of the matrix $D^{(k)}$, while this vertex is not available as internal vertex in the paths whose lengths are captured in $D^{(k-1)}$. If no v_i - v_j path utilising the first k vertices v_1, \dots, v_k of G is shorter than any v_i - v_j path utilising only the first $k-1$ vertices v_1, \dots, v_{k-1} of G , then it follows that $d_{ij}^{(k)} = d_{ij}^{(k-1)}$. However, if there is a v_i - v_j path \mathcal{P} utilising only the first k vertices v_1, \dots, v_k that is shorter than any path v_i - v_j utilising only the first $k-1$ vertices v_1, \dots, v_{k-1} of G , then $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$. Combining the above two cases above, it is possible to construct the (i, j) -th element of the matrix $D^{(k)}$ using the recursive relationship

$$d_{ij}^{(k)} = \min \left\{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \right\}, \quad k = 1, \dots, n. \quad (2.1)$$

The weight matrix $W(G) = D^{(0)}$ is thus used as initial matrix in a sequence of matrices $D^{(0)}, \dots, D^{(n)}$ generated by Floyd's algorithm.

Algorithm 2.1: Floyd's Algorithm

Input : The weight matrix $D^{(0)} = [d_{ij}^{(0)}]$ of a graph of order n .

Output: A matrix $D^{(n)} = [d_{ij}^{(n)}]$ of shortest distances between all pairs of vertices of G .

```

1 for  $k = 1$  to  $n$  do
2   for  $i = 1$  to  $n$  do
3     for  $j = 1$  to  $n$  do
4        $d_{ij}^{(k)} = \min \left\{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \right\};$ 
5 return  $[d_{ij}^{(n)}];$ 

```

The sequence of matrices $D^{(0)}, \dots, D^{(5)}$ computed by means of Algorithm 2.1 for the graph G_6 in Figure 2.7(a) is given by

$$\begin{aligned}
 D^{(0)} &= \begin{bmatrix} 0 & 1 & \infty & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty \\ \infty & 1 & 0 & 1 & 1 \\ \infty & \infty & 1 & 0 & 1 \\ \infty & \infty & 1 & 1 & 0 \end{bmatrix}, & D^{(1)} &= \begin{bmatrix} 0 & 1 & \infty & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty \\ \infty & 1 & 0 & 1 & 1 \\ \infty & \infty & 1 & 0 & 1 \\ \infty & \infty & 1 & 1 & 0 \end{bmatrix}, \\
 D^{(2)} &= \begin{bmatrix} 0 & 1 & 2 & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty \\ 2 & 1 & 0 & 1 & 1 \\ \infty & \infty & 1 & 0 & 1 \\ \infty & \infty & 1 & 1 & 0 \end{bmatrix}, & D^{(3)} &= \begin{bmatrix} 0 & 1 & 2 & 3 & 3 \\ 1 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 1 & 0 \end{bmatrix}, \\
 D^{(4)} &= \begin{bmatrix} 0 & 1 & 2 & 3 & 3 \\ 1 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 1 & 0 \end{bmatrix} & \text{and } D^{(5)} &= \begin{bmatrix} 0 & 1 & 2 & 3 & 3 \\ 1 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 1 & 0 \end{bmatrix}.
 \end{aligned}$$

2.1.4 Graph operations and special graphs

A graph of order n that consists only of a path is called a *path graph* and is denoted by \mathcal{P}_n (see Figure 2.8 for graphical representations of the paths \mathcal{P}_i for $i \in \{1, \dots, 5\}$), whereas a graph of order n that only consists of a cycle is called a *cycle graph* and is denoted by \mathcal{C}_n . Figure 2.9 contains graphical representations of the cycles \mathcal{C}_i for $i \in \{3, \dots, 7\}$.



FIGURE 2.8: Path Graphs.

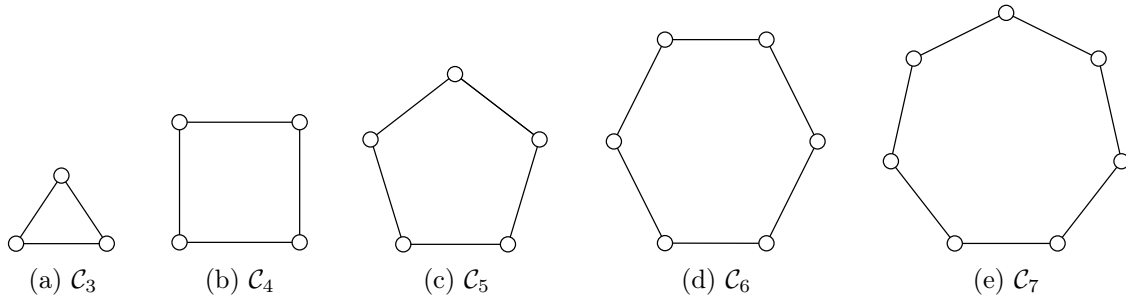


FIGURE 2.9: Cycle Graphs.

Operations may be performed on graphs to form other graphs. For example, the *union* of two graphs H_1 and H_2 , denoted by $H = H_1 \cup H_2$, is the graph H with vertex set $V(H) = V(H_1) \cup V(H_2)$ and edge set $E(H) = E(H_1) \cup E(H_2)$. The union of ℓ isomorphic copies of the graph G is denoted by ℓG . The *join* $H' = H_1 + H_2$ of two graphs H_1 and H_2 is the graph H' with vertex set $V(H') = V(H_1) \cup V(H_2)$ and edge set $E(H') = E(H_1) \cup E(H_2) \cup \{uv \mid u \in V(H_1) \text{ and } v \in V(H_2)\}$. The *cartesian product* $H'' = H_1 \square H_2$ of two graphs H_1 and H_2 is the graph H'' with vertex set $V(H'') = V(H_1) \times V(H_2)$ and in which two vertices (u_1, u_2) and (v_1, v_2) are adjacent if and only if either

$$u_1 = v_1 \text{ and } u_2 v_2 \in E(H_2) \quad \text{or} \quad u_2 = v_2 \text{ and } u_1 v_1 \in E(H_1).$$

These concepts are illustrated in Figure 2.10 for the graphs $H_1 = \mathcal{C}_3$ and $H_2 = \mathcal{P}_4$.

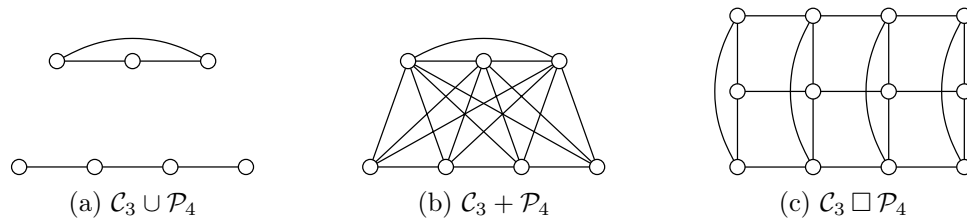
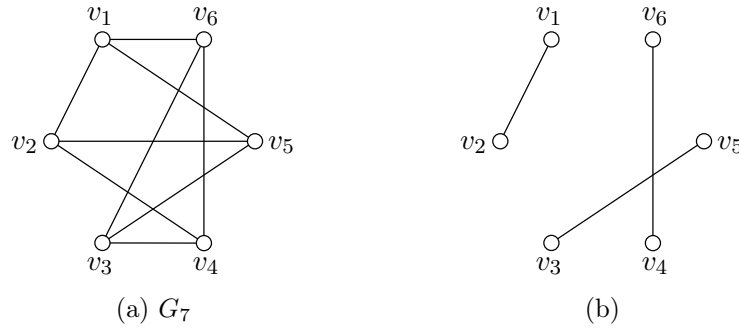


FIGURE 2.10: (a) The union of the graphs \mathcal{C}_3 and \mathcal{P}_4 . (b) The join of the graphs \mathcal{C}_3 and \mathcal{P}_4 . (c) The cartesian product of the graphs \mathcal{C}_3 and \mathcal{P}_4 .

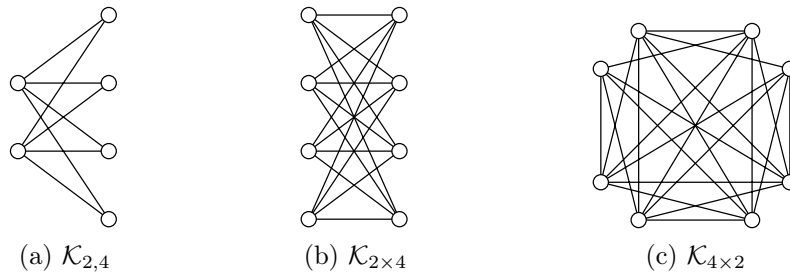
A graph G is *r-regular* if the degree of every vertex in G is r and a graph is *regular* if it is *r-regular* for some $r \in \mathbb{N}_0$. Any 1-regular subgraph of G is called a *matching* of G . A matching of G of the largest possible order is called a *maximum matching* of G , and the *matching number* of G , denoted by $\nu(G)$, is the cardinality of a maximum matching of G . A *perfect matching* of G is a matching containing all the vertices of G , if such a matching exists. The 3-regular graph G_7 in Figure 2.11(a) possesses a perfect matching, shown in Figure 2.11(b).


 FIGURE 2.11: (a) A 3-regular graph, G_7 . (b) A perfect matching of G_7 .

A *complete graph* is a graph in which every two distinct vertices are adjacent. A complete graph of order n is denoted by \mathcal{K}_n . The complete graph \mathcal{K}_n of order n is therefore a $(n - 1)$ -regular and has size $\binom{n}{2}$. Figure 2.6(a) contains a graphical illustration of a complete graph of order 4, (i.e. $G_5 \cong \mathcal{K}_4$). In contrast to a complete graph, the graph $\overline{\mathcal{K}_n}$ is called the *edgeless graph* of order n , since $E(\overline{\mathcal{K}_n}) = \emptyset$.

A graph G is k -*partite* for some natural number $k \geq 2$, if it is possible to partition the vertex set $V(G)$ into k nonempty subsets V_1, \dots, V_k , called *partite sets*, in such a way that no pair of vertices of V_i are adjacent, for all $i = 1, \dots, k$. If $k = 2$, the graph G is called *bipartite*, while if $k > 2$, G is called *multipartite*. If every vertex in any partite set V_i of a k -partite graph is adjacent to every vertex in $v \in V(G) - V_i$, then the graph is called a *complete k -partite graph* and is denoted by $\mathcal{K}_{n_1, \dots, n_k}$, where $|V_i| = n_i$ for all $i = 1, \dots, k$. The graph $\mathcal{K}_{n_1, \dots, n_k}$ may also be constructed as the join of k edgeless graphs, i.e. $\overline{\mathcal{K}_{n_1}} + \dots + \overline{\mathcal{K}_{n_k}}$. Furthermore, the complete bipartite graph $\mathcal{K}_{1, n-1}$ is a popular graph, called an n -star. A *galaxy* is a forest in which each component is a star, that is each component of the graph is a bipartite graph of the form $\mathcal{K}_{1, n}$. If $n_1, \dots, n_k = N$ (say), then the graph is called a *complete balanced k -partite graph* and is denoted $\mathcal{K}_{k \times N}$. A graphical illustration of the complete bipartite graph $\mathcal{K}_{2,4}$ is shown in Figure 2.12(a), while graphical illustrations of the complete multipartite graph $\mathcal{K}_{2 \times 4}$ and the complete balanced multipartite graph $\mathcal{K}_{4 \times 2}$ are shown in Figure 2.12(b) and (c), respectively.

A *planar graph* is a graph that can be embedded in the plane, i.e. drawn in the plane in such a way that its edges intersect only at their endpoints, letting no edges cross each other internally. Notice that the graph $\mathcal{K}_{2,4}$ in Figure 2.12(a) is planar (although in Figure 2.12(a) it is not presented without internal edge crossings), while $\mathcal{K}_{4 \times 2}$ in Figure 2.12(c) is not planar.


 FIGURE 2.12: Examples of complete graphs. (a) The complete bipartite graph $\mathcal{K}_{2,4}$. (b) The complete multipartite graph $\mathcal{K}_{2 \times 4}$. (c) The complete balanced multipartite graph $\mathcal{K}_{4 \times 2}$.

Let G_1, \dots, G_t be t nonempty graphs with $V(G_1) = \dots = V(G_t) = V(G)$. If it holds, for any edge $e \in E(G_i)$, that $e \notin E(G_j)$ for all $j \in \{1, \dots, t\}$ and $j \neq i$, the graphs G_1, \dots, G_t are called

pairwise edge-disjoint. The edge union of such graphs, denoted by

$$G = \bigoplus_{i=1}^t G_i = G_1 \oplus \cdots \oplus G_t, \quad (2.2)$$

is the graph with vertex set $V(G)$ and edge set $E(G) = \cup_{i=1}^t E(G_i)$. Each graph in (2.2) is called a *factor* of G , and G is said to be *factorable* into the factors G_1, \dots, G_t . The entire expression in (2.2) is called a *factorisation* of G . The graph $\mathcal{C}_3 \cup \mathcal{P}_4$ in Figure 2.10(a) is a factor of the graph $\mathcal{C}_3 + \mathcal{P}_4$ in Figure 2.10(b), while a factorisation of $\mathcal{C}_3 + \mathcal{P}_4$ is $(\mathcal{C}_3 \cup \mathcal{P}_4) \oplus \mathcal{K}_{3,4}$.

Let $n \geq 1$ be an integer and let $1 \leq i_1, \dots, i_z \leq n-1$ be z distinct integers. Then the *circulant* $\mathcal{C}_n\langle i_1, \dots, i_z \rangle$ of order n is a graph with vertex set $V(\mathcal{C}_n\langle i_1, \dots, i_z \rangle) = \{v_0, \dots, v_{n-1}\}$ and edge set $E(\mathcal{C}_n\langle i_1, \dots, i_z \rangle) = \{v_\alpha v_{(\alpha+\beta) \pmod n} \mid \alpha \in \{0, \dots, n-1\} \text{ and } \beta \in \{i_1, \dots, i_z\}\}$, where the notation $a \pmod b$ denotes the remainder when a is divided by b . The set $\{i_1, \dots, i_z\}$ is called the *connection set* of the circulant $\mathcal{C}_n\langle i_1, \dots, i_z \rangle$. Note that the complete graph \mathcal{K}_n is therefore the circulant $\mathcal{C}_n\langle 1, \dots, \lfloor n/2 \rfloor \rangle$, while the empty graph $\bar{\mathcal{K}}_n$ is therefore the circulant of order n with an empty connection set. If $z = 1$ the circulant $\mathcal{C}_n\langle i \rangle$ is said to be *elementary*, else it is called *composite*. If n is even and $i = \frac{n}{2}$, then $\mathcal{C}_n\langle i \rangle$ is called a *singular* (elementary) circulant. A composite circulant $\mathcal{C}_n\langle i_1, \dots, i_z \rangle$ is singular if $i_j = \frac{n}{2}$ for some $j \in \{1, \dots, z\}$. A composite circulant may therefore be constructed from two or more elementary circulants and is written as the edge union $\mathcal{C}_n\langle i_1, \dots, i_z \rangle = \bigoplus_{s=1}^z \mathcal{C}_n\langle i_s \rangle$ of elementary circulants. Figure 2.13(a) contains a graphical representation of the composite circulant $\mathcal{C}_8\langle 2, 3, 4 \rangle$, together with its factors $\mathcal{C}_8\langle 2 \rangle$, $\mathcal{C}_8\langle 3 \rangle$ and $\mathcal{C}_8\langle 4 \rangle$ in Figure 2.13(b)–(d).

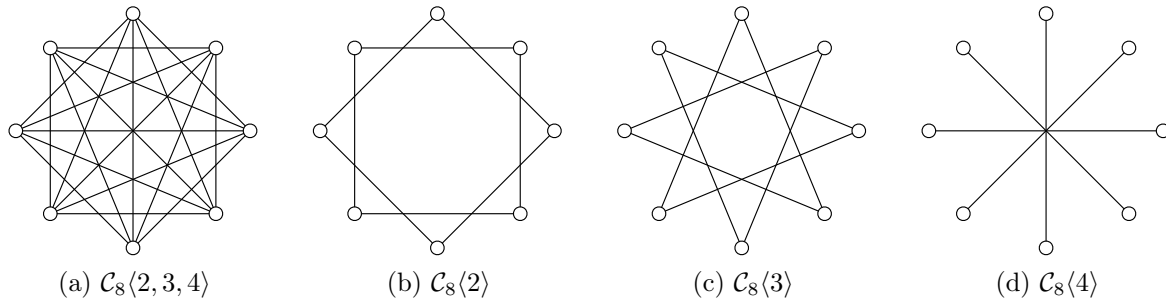


FIGURE 2.13: The graph (a) $\mathcal{C}_8\langle 2, 3, 4 \rangle$ together with its factors (b) $\mathcal{C}_8\langle 2 \rangle$, (c) $\mathcal{C}_8\langle 3 \rangle$ and (d) $\mathcal{C}_8\langle 4 \rangle$.

Let G be a graph of order n with vertex set $\{v_1, \dots, v_n\}$ and let $S = \{u_1, \dots, u_n\}$ be a set of vertices disjoint from V . The *corona* of G , denoted by $G \circ \mathcal{K}_1$, is the graph with vertex set $V(G) \cup S$ and edge set $E(G) \cup \{v_i u_i \mid i = 1, \dots, n\}$. Stated informally, the corona of a graph G is the graph that is obtained by joining a new pendent vertex to each vertex of G . A graphical illustration of the corona of $\mathcal{K}_{2,4}$ is shown in Figure 2.14(a).

Consider a cycle \mathcal{C}_n of length $n \geq 3$ with vertex set $\{v_1, \dots, v_n\}$, and another vertex v_0 (say). The *wheel* \mathcal{W}_n of order $n+1$ is the graph join $\mathcal{C}_n + \{v_0\}$, with the vertex v_0 sometimes referred to as the *hub*. The edges joining the hub to the rest of the graph are often referred to as *spokes*. A graphical illustration of the wheel \mathcal{W}_6 is shown in Figure 2.14(b).

A *spider*, denoted by $\mathcal{S}(M \times N)$, is a tree consisting of M paths, each of order N , intersecting at a single end-vertex. If the paths in a similarly constructed graph are not all of the same length, the graph is called a *wounded spider* and is denoted by $\mathcal{S}(n_1, n_2, \dots, n_k)$, where $n_i \geq 1$ denotes the order of the i^{th} path, for $i = 1, \dots, k$. A graphical illustration of the spider $\mathcal{S}(4 \times 3)$ is shown in Figure 2.14(c), while a graphical illustration of the wounded spider $\mathcal{S}(1, 3, 2, 3)$ is shown in Figure 2.14(d).

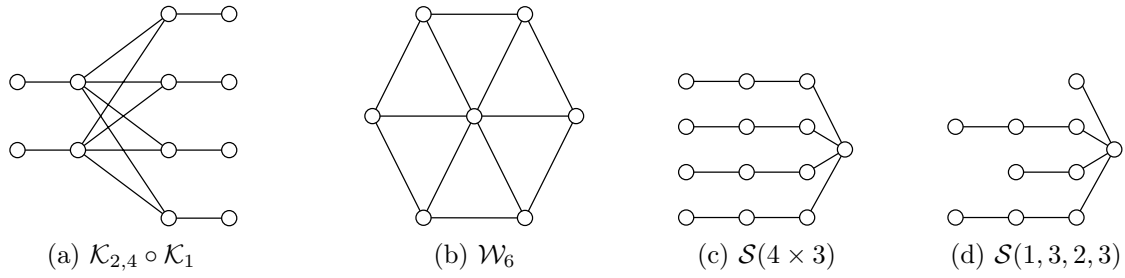


FIGURE 2.14: (a) The corona of $K_{2,4}$, (b) the wheel W_6 , (c) the spider $S(4 \times 3)$ and (d) the wounded spider $S(1, 3, 2, 3)$.

Finally, if $p, q \in \mathbb{N}$, the *hexagonal graph* $\mathcal{H}_{p,q}$ is the union of the cartesian product $\mathcal{P}_p \square \mathcal{P}_q$, with the edge sets $\{v_{2i,j}v_{2i-1,j+1} \mid i = 1, \dots, \lceil \frac{p}{2} \rceil \text{ and } j = 1, \dots, q-1\}$ and $\{v_{2i,j-1}v_{2i+1,j} \mid i = 2, \dots, \lceil \frac{p}{2} \rceil - 1 \text{ and } j = 2, \dots, q\}$. An illustration of such a graph is shown in Figure 2.15.

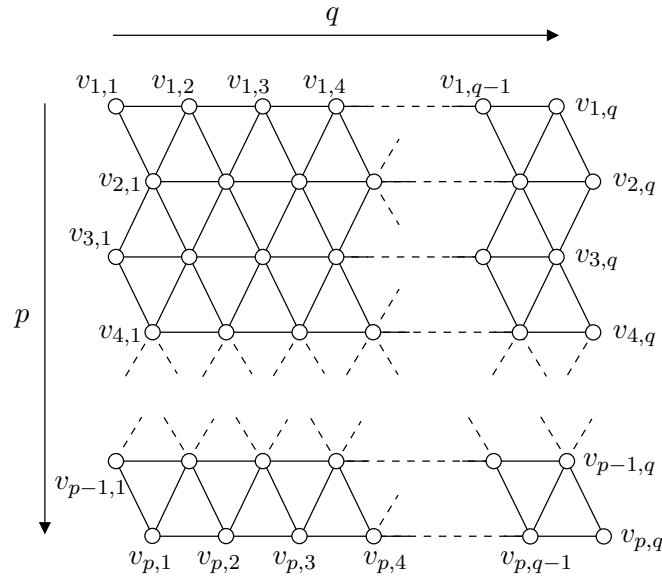


FIGURE 2.15: Graphical representation of the hexagonal graph $\mathcal{H}_{p,q}$.

An assignment of colours to the vertices of a graph G , one colour to each vertex, so that adjacent vertices are assigned different colours is called a *proper colouring* of G . A proper colouring in which k colours are used is called a *k -proper colouring*. The minimum value of k for which a graph G has a k -proper colouring is the *chromatic number* of G and is denoted by $\chi(G)$.

Brooks [11] provided the following bound on the chromatic number of a graph.

Theorem 2.2 (Brooks) *For any graph G which is neither a complete graph nor an odd cycle, $\chi(G) \leq \Delta(G)$.* ■

The bound in Theorem 2.2 is sharp; it is attained by an even cycle. The chromatic number of any graph G may also be bounded from above in terms of its size, as shown by Mitchell [77].

Theorem 2.3 (Mitchell) *For any graph G of size m , $\chi(G) \leq \frac{1}{2}(1 + \sqrt{8m+1})$.* ■

The bound in Theorem 2.3 is sharp; it is attained by a complete graph.

2.2 Fundamentals of graph domination

Let G be a graph with vertex set $V(G)$ and let P be a property of sets. A set $S \subseteq V(G)$ is called a P -set if it satisfies property P ; otherwise, S is a \bar{P} -set. A P -set S is *maximal* if every proper superset $S' \supset S$ is a \bar{P} -set. Similarly, a P -set S is *minimal* if every proper subset $S' \subset S$ is a \bar{P} -set. A set of vertices S with property P is *hereditary* if every proper subset $S' \subset S$ also has property P . Similarly, a set of vertices S with property P is *superhereditary* if every proper superset $S' \supset S$ also has property P .

2.2.1 Domination, independence and irredundance

A set $S \subseteq V(G)$ of vertices is a *dominating set* of a graph G if every vertex $v \in V(G)$ is an element of S or is adjacent to an element of S , that is, if $N[S] = V(G)$. Furthermore, any set $S \subseteq V(G)$ is said to *dominate* a vertex $v \in V(G)$ if $v \in N[S]$. A dominating set S of G is *minimal* if $S - v$ is not a dominating set of G for any $v \in S$. Since every superset of a dominating set of a graph is again a dominating set of the graph, domination is a superhereditary property. The *domination number* of a graph G , denoted by $\gamma(G)$, and the *upper domination number* of G , denoted by $\Gamma(G)$, are respectively the minimum and the maximum cardinalities of minimal dominating sets of G .

The set $S = \{v_3, v_4\}$, denoted by solid vertices in Figure 2.16(b), is a minimal dominating set of minimum cardinality for the graph G_8 . The set of vertices S , denoted by the solid vertices in Figure 2.16(c), is a minimal dominating set of maximum cardinality for the graph G_8 . Furthermore, the set of solid vertices in Figure 2.16(d) is also a minimal dominating set of G_8 . It follows that $\gamma(G_8) = 2$ and that $\Gamma(G_8) = 4$.

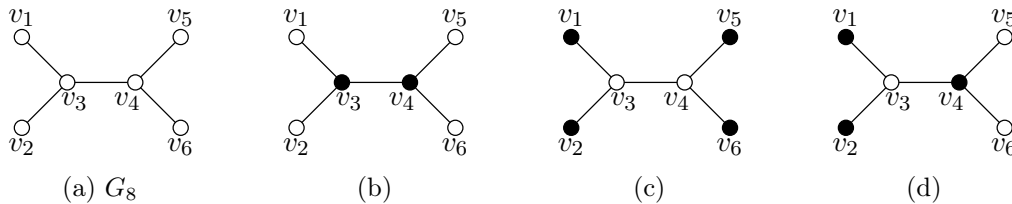


FIGURE 2.16: (a) A graph G_8 . (b) A dominating set of minimum cardinality for the graph G_8 . (c) An example of a minimal dominating set of cardinality 4 for G_8 , which is also a maximum cardinality minimal dominating set as well as a maximum cardinality maximal independent set. (d) An example of a minimal dominating set of cardinality 3 for G_8 , which is also a minimum cardinality maximal independent set.

A set $S \subseteq V(G)$ is an *independent set* of a graph G if the induced subgraph $\langle S \rangle_G$ contains no edges, that is, if $N(v) \cap S = \emptyset$ for every $v \in S$. An independent set of G is *maximal* if the set $S \cup \{u\}$ is not independent for any $u \in V(G) - S$. The minimum and maximum cardinalities of a maximal independent set are called respectively the *lower independence number*, denoted by $i(G)$, and the *independence number*, denoted by $\beta(G)$, of G . Since every subset of an independent set of a graph G is again an independent set of G , independence is a hereditary property.

A maximum cardinality maximal independent set of the graph G_8 in Figure 2.16(a) is shown in Figure 2.16(c), while a minimum cardinality maximal independence set of the same graph is shown in Figure 2.16(d). This shows that $i(G_8) = 3$ and that $\beta(G_8) = 4$.

Berge [6] was first to observe that if S is an independent set of a graph G and $v \in V(G) - S$, the set $S \cup \{v\}$ is not independent in G if and only if S dominates v .

Theorem 2.4 (Berge [6]) *Let G be any graph.*

- (a) *An independent set of G is maximal independent if and only if it is both an independent set and a dominating set of G .*
- (b) *Every maximal independent set of G is a minimal dominating set of G .* ■

The following corollary follows immediately from Theorem 2.4.

Corollary 2.2 (Haynes *et al.* [60]) *For any graph G ,*

$$\gamma(G) \leq i(G) \leq \beta(G) \leq \Gamma(G).$$

A vertex u is called a *private neighbour* of v with respect to S if $N[u] \cap S = \{v\}$. Furthermore, the *private neighbourhood* of v relative to S is the set $N[v] - N[S - \{v\}]$ and is denoted by $\text{pn}(v, S)$. The set of *external private neighbours* of v relative to S are the vertices in the set $\text{pn}(v, S) - \{v\}$, which is denoted by $\text{Epn}(v, S)$. The *internal private neighbours* of v relative to S are the vertices in the set $\{u \in S \mid N(u) \cap S = \{v\}\}$, which is denoted by $\text{Ipn}(v, S)$.

Consider the set $S = \{v_3, v_5, v_6\}$ in the graph G_8 in Figure 2.16(a). For this set S , it follows that $\text{pn}(v_3, S) = \{v_1, v_2, v_3\}$, $\text{Epn}(v_3, S) = \{v_1, v_2\}$ and $\text{Ipn}(v_3, S) = \emptyset$.

A set $S \subseteq V(G)$ is an *irredundant* set of a graph G if every vertex $v \in S$ has at least one private neighbour, *i.e.* for every vertex $v \in S$, $\text{pn}(v, S) \neq \emptyset$. A vertex v in some subset $S \subseteq V(G)$ is an *irredundant vertex* of S if $\text{pn}(v, S) \neq \emptyset$; otherwise it is a *redundant vertex* of S .

Cockayne *et al.* [35, Proposition 4.2] were the first to note the following relationship between domination and irredundance.

Proposition 2.1 (Cockayne *et al.* [35]) *For any graph G , a set S is dominating and irredundant if and only if it is a minimal dominating set of G .* ■

As with independence, irredundance is a hereditary property. The *lower irredundance number*, denoted by $ir(G)$, and the *irredundance number*, denoted by $IR(G)$ are respectively the minimum and the maximum cardinalities of the maximal irredundant sets of G . A minimum cardinality maximal irredundant set of the graph G_9 in Figure 2.17(a) is shown in Figure 2.17(b), while a maximum cardinality maximal irredundant set of the same graph is shown in Figure 2.17(c). This shows that $ir(G_9) = 2$ and that $IR(G_9) = 4$.

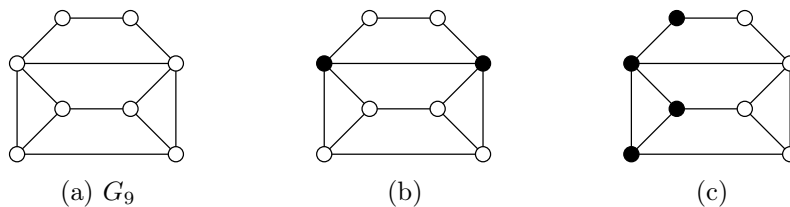


FIGURE 2.17: (a) A graph G_9 . (b) A maximal irredundant set of minimum cardinality of the graph G_9 . (c) A maximal irredundant set of maximum cardinality of the graph G_9 .

Haynes *et al.* [60] credited the following result to Bollobás and Cockayne [7], while Cockayne and Hedetniemi [34] were, in fact, the actual discoverers of the result.

Proposition 2.2 (Cockayne & Hedetniemi [34]) *Every minimal dominating set of a graph G is a maximal irredundant set of G .* ■

As a result of Proposition 2.2, the following inequality chain was first observed by Cockayne, Hedetniemi and Miller in 1978.

Theorem 2.5 (Cockayne *et al.* [35]) *For any graph G ,*

$$ir(G) \leq \gamma(G) \leq i(G) \leq \beta(G) \leq \Gamma(G) \leq IR(G).$$

These six parameters in Theorem 2.5 are often collectively referred to as the *domination parameters* of G . Furthermore, ir , γ and i are usually called the *lower domination parameters* and β , Γ and IR are usually called the *upper domination parameters*.

A survey of the literature on classical domination is conducted in the following subsections. A number of related domination parameters are also introduced.

2.2.2 The domination number of a graph

In 1962 Ore [80, Theorems 13.1.1, 13.1.3, 13.1.4 and 13.1.5] provided the following fundamental properties of minimal dominating sets.

Theorem 2.6 (Ore [80]) *Let G be any graph. Then*

- (a) *any dominating set of G contains a minimal dominating set of G .*
- (b) *a dominating set $S \subseteq V(G)$ is minimal if and only if, for every vertex $v \in S$, v is an isolate of S or there exists a vertex $u \in V(G) - S$ for which $N(u) \cap S = \{v\}$.*

Furthermore, if G has no isolated vertices, then

- (c) *the complement of a minimal dominating set of G is a dominating set.*
- (d) *G contains at least two disjoint minimal dominating sets.* ■

Bollobás and Cockayne [7] established the following additional property of minimum dominating sets in graphs in 1979.

Theorem 2.7 (Bollobás & Cockayne [7]) *If G is a graph without isolated vertices, then there exists a dominating set X in which $\text{Epn}(v, X) \neq \emptyset$ for every vertex $v \in X$.* ■

It is obvious that at least one vertex is required to dominate a graph, and that the domination number of a graph is bounded from above by its order (*i.e.* $1 \leq \gamma(G) \leq n$ for any graph of order n). The lower bound is attained by the complete graph K_n , while the upper bound is attained by the empty graph \overline{K}_n . The following improvement of the upper bound for isolate-free graphs dates back to 1962 and is a direct consequence of the result of Theorem 2.6(c).

Corollary 2.3 (Ore [80]) *For any graph G of order n without isolated vertices, $\gamma(G) \leq \frac{n}{2}$.* ■

Fink *et al.* [43, Theorem 3] and Payan and Xuong [83, pp. 24] independently characterised the class of isolate-free graphs with domination number exactly half their order as follows.

Theorem 2.8 (Fink *et al.* [43] and Payan & Xuong [83]) *For any connected graph G of order $2n$ without isolated vertices, $\gamma(G) = n$ if and only if $G \cong C_4$ or $G \cong H \circ K_1$, where H is any connected graph.* ■

Whereas Ore's bound in Corollary 2.3 holds for any graph with minimum degree at least one, McCuaig and Shepherd [76, Theorem 3] were able to improve on this bound for graphs with minimum degree at least two.

Theorem 2.9 (McCuaig & Shepherd [76]) *For any connected graph $G \notin \mathcal{A}$ of order n without isolated or end-vertices, $\gamma(G) \leq \frac{2n}{5}$, where the set \mathcal{A} of forbidden graphs appears in Figure 2.18.* ■

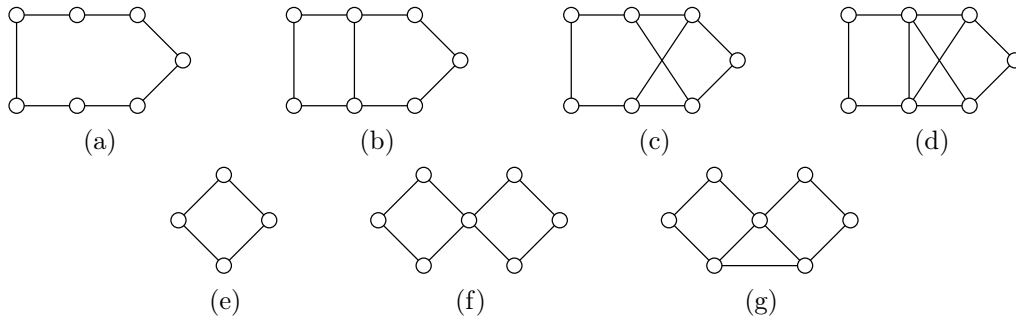


FIGURE 2.18: The family \mathcal{A} of forbidden graphs in Theorem 2.9.

If the minimum degree is further increased to be at least three, then the upper bound in Theorem 2.9 can be improved. Reed [86] achieved such an improvement in 1996.

Theorem 2.10 (Reed [86]) *For any connected graph G of order n with minimum degree at least 3, $\gamma(G) \leq \frac{3n}{8}$.* ■

Harant *et al.* [56] established the following bound in 1999 for any graph in terms of its minimum degree.

Theorem 2.11 (Harant *et al.* [56]) *For any graph G of order n with minimum degree δ ,*

$$\gamma(G) \leq n \left[1 - \delta \left(\frac{1}{1 + \delta} \right)^{1 + \frac{1}{\delta}} \right].$$

Arnaoutov [2] proved the following upper bound on the domination number of a graph in terms of its order and minimum degree in 1974.

Theorem 2.12 (Arnaoutov [2]) *For any graph G of order n with minimum degree $\delta \geq 1$,*

$$\gamma(G) \leq \frac{n}{\delta + 1} \sum_{j=1}^{\delta+1} \frac{1}{j}.$$

The following bound was proved by Alon and Spencer [1] in 1992 using the probabilistic method. The probabilistic method creates a subset of the vertices X of an isolate-free graph G , by selecting each vertex $v \in V(G)$ with a uniform probability. A set of vertices Y are identified in $V(G) - X$ without any neighbours in X . By definition it follows that $X \cup Y$ is a dominating set of G . They prove that the expected cardinality of $X \cup Y$ is bounded from above.

Theorem 2.13 (Alon & Spencer [1]) *For any isolate-free graph G of order n with minimum degree δ ,*

$$\gamma(G) \leq \frac{n[1 + \ln(\delta + 1)]}{\delta + 1}. \quad \blacksquare$$

The maximum degree of a graph may also be used to express bounds on the domination number of a graph. The lower bound in the following theorem is due to Walikar, Acharya and Sampathkumar [103] and dates back to 1979, while the upper bound is due to Berge [6] and dates back to 1962.

Theorem 2.14 (Berge [6] and Walikar *et al.* [103]) *For any graph G of order n with maximum degree Δ ,*

$$\left\lceil \frac{n}{1 + \Delta} \right\rceil \leq \gamma(G) \leq n - \Delta. \quad \blacksquare$$

Both bounds in Theorem 2.14 are attained by a galaxy. In 1995 Slater [90] established the following lower bound on the domination number of a graph in terms of its degree sequence.

Theorem 2.15 (Slater [90]) *For any graph G of order n with non-increasing degree sequence d_1, \dots, d_n ,*

$$\gamma(G) \geq \min_k \{k + (d_1 + d_2 + \dots + d_k) \geq n\}. \quad \blacksquare$$

Flach and Volkmann [45, Theorem 3] established the following upper bound in 1990 on the domination number of a graph in terms of its order and both its minimum and maximum degrees.

Theorem 2.16 (Flach & Volkmann [45]) *For any graph G of order n with minimum degree δ and maximum degree Δ ,*

$$\gamma(G) \leq \left(n + 1 - \Delta \frac{\delta - 1}{\delta} \right) / 2. \quad \blacksquare$$

The following corollary to Theorem 2.16 for isolate-free graphs is due to Payan [82] and dates from 1975.

Corollary 2.4 (Payan [82]) *For any graph G without isolated vertices,*

$$\gamma(G) \leq \frac{n + 2 - \delta(G)}{2}. \quad \blacksquare$$

A number of bounds on the domination number of a graph have also been established in terms of the order and size of the graph. The following bound in this class was established in 1965.

Theorem 2.17 (Vizing [100]) *For any graph G of order n and size m without universal vertices,*

$$m \leq \left\lfloor \frac{1}{2}(n - \gamma(G))(n + 2 - \gamma(G)) \right\rfloor. \quad \blacksquare$$

In 1998 Haynes *et al.* [60, Theorem 2.22] established the following bounds on $\gamma(G)$, again in terms of the order and size of G .

Theorem 2.18 (Haynes *et al.* [60]) *For any graph G of order n and size m ,*

$$n - m \leq \gamma(G) \leq n + 1 - \sqrt{1 + 2m}.$$

Furthermore, $\gamma(G) = n - m$ if and only if G is a galaxy. \blacksquare

Bounds on the domination number of a graph may also be formulated in terms of the diameter and girth of the graph. A graph G with a diameter of 2 contains a dominating set $S = N(v)$, where v is any vertex in $V(G)$. The following upper bound follows immediately from this observation.

Theorem 2.19 *For any graph G with diameter equal to 2, $\gamma(G) \leq \delta(G)$.* \blacksquare

Another elementary bound on the domination number of a graph in terms of its diameter is due to Haynes *et al.* [60] and it dates from 1998.

Theorem 2.20 (Haynes *et al.* [60]) *For any connected graph G ,*

$$\left\lceil \frac{\text{diam}(G) + 1}{3} \right\rceil \leq \gamma(G). \quad \blacksquare$$

Brigham *et al.* [9] provided the following relationship between the diameter of a graph and the domination number of its complement.

Theorem 2.21 (Brigham *et al.* [9]) *If, for any graph G , $\gamma(\overline{G}) \geq 3$, then $\text{diam}(G) \leq 2$.* \blacksquare

Brigham and Dutton [10] obtained a number of bounds on the domination number of a graph in terms of its minimum degree and girth, as summarised in the following theorem.

Theorem 2.22 (Brigham & Dutton [10]) *Let G be any graph with minimum degree δ and girth g .*

$$(a) \text{ If } g \geq 5, \text{ then } \delta \leq \gamma(G) \leq \left\lceil \frac{n - \lfloor g/3 \rfloor}{2} \right\rceil.$$

$$(b) \text{ If } g \geq 6, \text{ then } \gamma(G) \geq 2(\delta - 1). \quad \blacksquare$$

The maximum degree of a graph also represents a lower bound on the domination number of a graph for graphs of large girth, as made more precise in the following theorem.

Theorem 2.23 (Brigham & Dutton [10]) *For any graph G with minimum degree $\delta \geq 2$, girth $g \geq 7$ and maximum degree Δ , $\gamma(G) \geq \Delta$.* \blacksquare

Finally, MacGillivray and Seyffarth [75] established the following upper bounds on the domination numbers of planar graphs with small diameters.

Theorem 2.24 (MacGillivray & Seyffarth [75]) *Let G be any planar graph.*

(a) *If $\text{diam}(G) = 2$, then $\gamma(G) \leq 3$.*

(b) *If $\text{diam}(G) = 3$, then $\gamma(G) \leq 10$.* ■

One of the most famous open problems involving the domination of a graph is Vizing's conjecture. In 1963 Vizing [99] posed the problem of finding a lower bound on the domination number of a graph G of the form $G \cong G_1 \square G_2$. He later conjectured as follows.

Conjecture 2.1 (Vizing's Conjecture [101]) *For any graphs G and H ,*

$$\gamma(G \square H) \geq \gamma(G) \cdot \gamma(H). \quad \blacksquare$$

Much of the subsequent literature related to Conjecture 2.1 involves showing that the conjecture holds for families of graphs satisfying specific conditions.

2.2.3 Multiple domination

In applications, a dominating set X of a graph G may often be interpreted as a set of vertices that either monitors or controls the vertices in $V(G) - X$. In such applications, it may be desirable to increase the level of domination of each vertex. A vertex in $V(G) - X$ is *k-dominated* if it is dominated by at least k vertices in X , that is, if $v \notin X$ then $|N(v) \cap X| \geq k$. If every vertex in $V(G) - X$ is k dominated, then X is called a *k-dominating set*. The minimum cardinality of a k -dominating set of a graph G is called the *k-domination number* of G and is denoted by $\gamma_k(G)$. A dominating set of G is therefore a special case of a k -dominating set of G , namely where $k = 1$, and so $\gamma(G) \leq \gamma_k(G)$ for any graph G and any natural number k .

Harary and Haynes [57], on the other hand, introduced the notion of *k-tuple domination*. A set $X \subseteq V(G)$ is a *k-tuple dominating set* of a graph G if each vertex in $V(G)$ is dominated by at least k vertices in X . The minimum cardinality of a k -tuple dominating set is called the *k-tuple domination number* and is denoted by $\gamma_{\times k}(G)$. A dominating set of G is therefore also a special case of a k -tuple dominating set of G , namely where $k = 1$, and so $\gamma(G) \leq \gamma_{\times k}(G)$ for any graph G and any natural number k .

The difference between k -domination and k -tuple domination is that in k domination the vertices in $V(G) - X$ are the only vertices of G that must be dominated multiple times, whereas in k -tuple domination *every* vertex of G must be dominated multiple times. It therefore follows that $\gamma_k(G) \leq \gamma_{\times k}(G)$ for any graph G and any natural number k .

2.2.4 Edge criticality in graph domination

An important consideration in the topological design of a network is fault tolerance, that is, the ability of a network to provide service even when it contains a faulty component. The behaviour of a network in the presence of a fault can be analysed by determining the effect on the tolerance of a network of removing an edge (representing a link failure) or a vertex (representing a processor

failure) from the underlying network or graph. For example, a dominating set of the underlying graph represents the minimum number of processors that are able to communicate directly with all the other processors in the network. It is important in the context of network fault tolerance to be able to determine when the domination number changes as a result of a vertex or edge being deleted from the graph. Similarly, the effects on the domination number can be analysed when an absent edge is added to the network, essentially improving on the fault-tolerance of the network.

Edge deletion and domination

It is evident that the removal of an edge from a graph G cannot decrease the value of the domination number and can increase this value by at most one [60]. A graph G for which the domination number changes when an arbitrary edge is removed from it is called a γ^+ -critical graph and has the property that $\gamma(G - e) = \gamma(G) + 1$ for all $e \in E(G)$. This class of graphs was characterised independently by Walikar and Acharya [102] in 1979 and Bauer *et al.* [3] in 1983.

Theorem 2.25 (Bauer *et al.* [3], Walikar & Acharya [102]) *A graph is γ^+ -critical if and only if it is a galaxy.* ■

In 1983 Bauer *et al.* [3] established the notion now known as the *bondage number* of a graph G , denoted by $b(G)$, as the minimum number of edge removals from G which ensures an increase the domination number of the resulting graph. Bauer *et al.* [3] originally called the bondage number of a graph its edge stability number and studied this graph parameter in the context of the so-called degrees of the edges of a graph. The *degree of an edge* uv of a graph G is $\deg_G(u) + \deg_G(v)$. Denote the smallest degree of any edge in G by $\delta'(G)$, and call this parameter the minimum edge degree of G .

Theorem 2.26 (Bauer *et al.* [3]) *Let G be any graph with minimum edge degree δ' and maximum degree Δ .*

- (a) *If there is at least one vertex $v \in V(G)$ for which $\gamma(G - v) \geq \gamma(G)$, then $b(G) \leq \Delta$.*
- (b) *If G is a nontrivial tree, then $b(G) \leq 2$.*
- (c) *$b(G) \leq \delta' - 1$.* ■

Fink *et al.* [44] studied the same concept in 1990 (in fact, they introduced the term “bondage number”). Unaware of the earlier work by Bauer *et al.* [3], they reproduced the same results published seven years earlier. They also established the bondage numbers for the infinite classes of paths, cycles and complete graphs as follows.

Theorem 2.27 (Fink *et al.* [44])

- (a) $b(K_n) = \lceil n/2 \rceil$.
- (b) $b(C_n) = \begin{cases} 3 & \text{if } n \equiv 1 \pmod{3} \\ 2 & \text{otherwise.} \end{cases}$
- (c) $b(P_n) = \begin{cases} 2 & \text{if } n \equiv 1 \pmod{3} \\ 1 & \text{otherwise.} \end{cases}$ ■

Fink *et al.* [44] also established the following bounds on the bondage number of an arbitrary graph.

Theorem 2.28 (Fink *et al.* [44]) *Let G be any graph of order n with minimum degree δ and maximum degree Δ .*

- (a) *If G is connected, then $b(G) \leq n - 1$.*
- (b) *If G is connected, then $b(G) \leq \Delta + \delta - 1$.*
- (c) *If $\gamma(G) \geq 2$, then $b(G) \leq (\gamma(G) - 1)\Delta + 1$.*
- (d) *If G is connected and $n \geq 2$, then $b(G) \leq n - \gamma(G) + 1$. ■*

The *edge connectivity* of a connected graph G , denoted by $\kappa_1(G)$, is the smallest number of edges whose removal from G disconnects the resulting graph. Hartnell and Rall [58] established the following bound on $b(G)$ in terms of its maximum degree and edge connectivity.

Theorem 2.29 (Hartnell & Rall [58]) *For any connected graph G with maximum degree Δ , $b(G) \leq \Delta + \kappa_1(G) - 1$. ■*

Fink *et al.* [44], in fact, conjectured that $b(G) \leq \Delta + 1$ for any non-empty graph G with maximum degree Δ . This bound is attained in Theorem 2.29 when $\kappa_1(G) = 2$. Chvátal and Cook [26] formulated an integer programming problem for determining the bondage number of a graph. Wang [104] improved upon the upper bound in Theorem 2.26(c), based on a classification of the vertices of a graph into four categories. Teschner [94] established various upper bounds on the bondage number of a graph, including the first lower bounds on the bondage number, and he was able to characterise trees with bondage number 1.

Edge addition and domination

In contrast to edge deletion, edge addition can only increase the domination number of a graph (by at most one). The class of graphs for which the domination number changes when an arbitrary edge is added to it is called γ^- -critical graphs. A γ^- -critical graph G therefore satisfies the property that $\gamma(G + e) = \gamma(G) - 1$ for any edge $e \in E(G)$. A significant amount of research has been done on domination criticality with respect to edge addition [59, Chapter 16]. Sumner and Blich [93], who used the term “edge domination critical graphs” when referring to γ^- -critical graphs, were able to characterise γ^- -critical graphs G as follows in the special cases where $\gamma(G) = 1$ or $\gamma(G) = 2$.

Theorem 2.30 (Sumner & Blich [93]) *For any γ^- -critical graph G ,*

- (a) *$\gamma(G) = 1$ if and only if $G \cong K_n$.*
- (b) *$\gamma(G) = 2$ if and only if \overline{G} is a galaxy. ■*

The problem of characterising γ^- -critical graphs with domination number at least 3 is much more difficult. Sumner [92], however, characterised the class of disconnected γ^- -critical graphs with domination number 3 as follows.

Theorem 2.31 (Sumner [92]) *A disconnected graph G with $\gamma(G) = 3$ is γ^- -critical if and only if $G \cong A \cup B$, where either A is trivial and B is a γ^- -critical graph with domination number 2, or A is a complete graph and B is a complete graph minus a 1-factor.* ■

Although the class of γ^- -critical graphs with domination number at least 3 has not been characterised fully, various properties of these graphs have been found. Favaron *et al.* [42], for example, were able to show that a γ^- -critical graph G with domination number $\gamma(G) = k$ has diameter at most $2k - 2$.

Theorem 2.32 (Sumner & Blich [93]) *If G is a γ^- -critical graph, then G contains no vertex for which $\gamma(G - v) > \gamma(G)$.* ■

The following property of the vertex set of a γ^- -critical graph dates back from 1994.

Theorem 2.33 (Favaron *et al.* [42]) *If G is a γ^- -critical graph, then all the vertices of G for which $\gamma(G - v) = \gamma(G)$ induce a complete subgraph of G .* ■

In 1990 Kok and Mynhardt [72] introduced the notion of the *reinforcement number* $r(G)$ of a graph G as the smallest number of arbitrary edges which must be added to G in order to decrease the domination number of the resulting graph. They characterised the reinforcement number for infinite families of graphs and also used the notion of the reinforcement to improve on the bound in Theorem 2.14.

Theorem 2.34 (Kok and Mynhardt [72]) *For any graph G of order n with maximum degree Δ and reinforcement number $r(G)$,*

$$\gamma(G) \leq n - \Delta - r(G) + 1. \quad \blacksquare$$

2.3 Basic notions of complexity theory

An algorithm may be defined as an ordered sequence of procedural operations for solving a problem within a finite number of operations [63]. Many computational problems in graph theory can be solved by algorithms¹. It is of vital importance to determine the efficiency of an algorithm in terms of the computational speed and the amount of computer memory required to execute the algorithm when solving increasingly larger instances of computational problems in graph theory. In deterministic algorithms each step is followed by a uniquely determined next step (such as in Floyd's algorithm), whereas one of several possibilities may be chosen randomly as the next step in nondeterministic algorithms [95].

Algorithmic complexity is the process of quantifying the number of basic operations performed and the amount of memory expended by a computer when performing the steps of an algorithm. Such a quantification is usually achieved by means of two variables: the *time complexity* $T(n)$ and the *space complexity* $S(n)$ of the algorithm, where n refers to the size of the input instance to the algorithm.

In Algorithm 2.1, the size n of the input instance to the algorithm may be taken as the order of the input graph G . Let a space unit be the amount of memory required to store the value of

¹An example of an algorithm that has already been encountered is Floyd's Algorithm, described in §2.1.3.

an integer in a computer. Then the amount of memory required to implement Algorithm 2.1 is $S(n) = 2n^2$ space units, because two distance matrices $D^{(k-1)}$ and $D^{(k)}$ must be stored during the k^{th} instance of the algorithm. Furthermore, the addition of two integers, variable assignment and the comparison of two integers may all be taken as basic (computational) operations in Algorithm 2.1. In Step 5, two integers are added, followed by the comparison of two integers and, lastly, one variable assignment is made. Since Step 5 is repeated n^3 times, the total number of variable assignments, variable comparisons and additions of two numbers is at most n^3 and so the time complexity of Algorithm 2.1 is $T(n) \leq 3n^3$.

Since Algorithm 2.1 is called only once (at the very start of the algorithm) during a single execution thereof and then iterates through a number of steps during execution, the algorithm is an example of an *iterative algorithm*, as opposed to a *recursive algorithm*. A recursive algorithm also iterates through a number of steps, but the algorithm may perform calls to itself during execution, called *recursive calls*. A smaller problem instance that is similar in nature and structure to the original problem instance being solved by the first call to a recursive algorithm, is typically solved during each recursive call.

The *worst-case* complexity of an algorithm is the largest possible values of $T(n)$ and $S(n)$ for any problem instance of input size n [48, p. 149]. Instead of performing exact counts of the computational resources required to execute a given algorithm, it is often sufficient to pursue a worst-case estimate of these resources, *i.e.* asymptotic upper bounds on the functions $S(n)$ and $T(n)$ as $n \rightarrow \infty$. The time [space, resp.] complexity of an algorithm may be determined by quantifying the amount of time [memory, resp.] expended by a computer when executing the instructions of an algorithm. For functions $f(\cdot)$ and $g(\cdot)$ mapping the set of positive integers to itself, $f(n) = \mathcal{O}(g(n))$ if there exist constants $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ [38, 105]. In this case the function $g(n)$ is an *asymptotic upper bound* for the function $f(n)$ as $n \rightarrow \infty$, and the function $f(n)$ is said to be of *order* the function $g(n)$. Since the time complexity $T(n)$ of Algorithm 2.1 satisfies $0 \leq T(n) \leq 3 \cdot n^3$ for all $n \geq 1$, as described above, it follows that this time complexity is $\mathcal{O}(n^3)$, while the space complexity of the algorithm is $\mathcal{O}(n^2)$ by a similar analysis.

An algorithm for which the time [space, resp.] complexity is asymptotically bounded from above by a linear function with respect to its input size n (*i.e.* which is $\mathcal{O}(n)$), is referred to as a *linear time* [space, resp.] *algorithm*, while the complexity is classified as *constant*, denoted $\mathcal{O}(1)$, if its complexity is independent of n . A *polynomial-time* [space, resp.] *algorithm* is an algorithm whose execution time [memory space required, resp.] is $\mathcal{O}(p(n))$ for some polynomial function $p(n)$; otherwise it is referred to as an *exponential time* [space, resp.] *algorithm*.

A computational problem is called *tractable* if it can be solved by a polynomial-time algorithm [38]; otherwise it is called *intractable*. It is, however, often difficult to classify a given computational problem as tractable or intractable. Complexity theory is the field of study in which tight asymptotic upper bounds are established on the computational resources required to solve computation problems, thereby classifying these problems as tractable or intractable.

2.3.1 The complexity classes P and NP

Decision theory is a branch of complexity theory which deals with the simplest forms of computational tasks — binary questions that may either be answered **true** or **false** [8, p. 175]. All decision problems that may be solved by polynomial-time algorithms are members of a class denoted by **P** (abbreviation for *Polynomial*). Therefore, a decision problem is in the class **P** if there exists an algorithm for solving any instance of size n of the problem in $\mathcal{O}(n^k)$ time for some

fixed integer k . The class **NP** (abbreviation for *Non-deterministic Polynomial*) comprises all decision problems that may be answered **true** by a polynomial-time algorithm, given additional information on the specific instance of the decision problem, called a *certificate*. Similarly, the class **co-NP** is the set of all decision problems that may be answered **false** by a polynomial-time algorithm, given additional information on the instance (a certificate). Although such certificates may exist for instances of a decision problem, finding these certificates may sometimes be difficult. Figure 2.19 is a graphical representation of the various classes of decision problems described in this section.

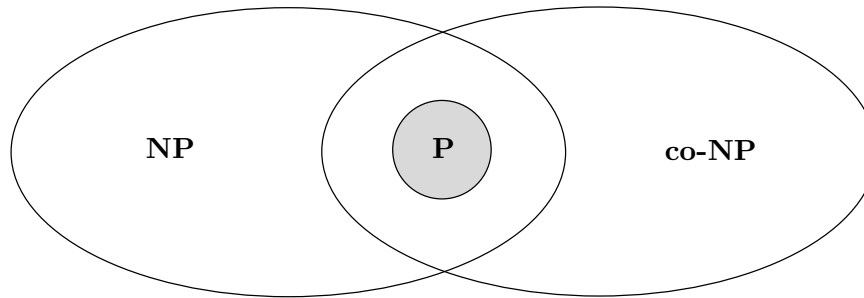


FIGURE 2.19: The complexity classes **P**, **NP** and **co-NP**.

Two fundamental open problems in decision theory are deciding whether or not $\mathbf{P} = \mathbf{NP}$ and whether or not $\mathbf{P} = \mathbf{co-NP}$, *i.e.* whether or not two or all of the sets in the Venn-diagram of Figure 2.19 coincide.

2.3.2 Reductions and NP-completeness

It is important to have a clear understanding of what it means for one problem to be at least as hard to solve as another. The notion of *reducibility* may be used to explain this concept. A decision problem D_2 *reduces* to a decision problem D_1 if there is a transformation R (often in the form of an algorithm) according to which every instance x of D_2 produces an equivalent instance $R(x)$ for D_1 [50, 81]. A decision problem D_1 is at least as hard to solve as decision problem D_2 if D_2 reduces in polynomial-time to D_1 , in which case D_2 is said to be polynomial-time reducible to D_1 .

Consider, as an example, the two decision problems

Decision Problem 2.1 (Connectedness)

Instance: A graph G of order n .

Question: Is G connected?

and

Decision Problem 2.2

Instance: A graph G of order n with vertex set $\{v_1, \dots, v_n\}$ and an integer $\ell \leq n$

Question: Is there a path between any pair of vertices of G utilizing a subset of only $\{v_1, \dots, v_\ell\}$ as internal vertices?

Decision Problem 2.2 may be solved in $\mathcal{O}(\ell n^2)$ time by computing the matrix $D^{(\ell)}$ in the sequence of matrices produced by Floyd's Algorithm, by allowing the outer index k in Algorithm 2.1 to

range from 1 to ℓ instead of from 1 to n and then testing in $\mathcal{O}(n^2)$ time whether all the entries of $D^{(\ell)}$ are finite. By taking $\ell = n$, it is clear that Decision Problem 2.1 is polynomial-time reducible to Decision Problem 2.2. Since Decision Problem 2.2 is in **P**, it therefore follows that Decision Problem 2.1 is also in **P**.

A decision problem D is **NP-hard** if any decision problem $D' \in \mathbf{NP}$ is polynomial-time reducible to D . Furthermore, a decision problem D is **NP-complete** if $D \in \mathbf{NP}$ and D is **NP-hard** [63]. An important observation is that an **NP-complete** decision problem D is a member of the class **P** if and only if $\mathbf{P} = \mathbf{NP}$. Similarly, if a decision problem D is **NP-complete** and $D \in \mathbf{co-NP}$, then $\mathbf{P} = \mathbf{co-NP}$. **NP-complete** problems may be considered the most restrictive subclass of **NP** decision problems within the above framework, as they are computationally at least as hard to solve as any other problem in **NP**. Figure 2.20 contains a graphical representation of the various classes of decision problems described in this and the previous sections. The next result provides a method for establishing the **NP-completeness** of a decision problem.

Theorem 2.35 *Let D_1 and D_2 be two decision problems. If $D_1 \in \mathbf{NP}$, D_2 is **NP-complete** and D_2 is polynomial-time reducible to D_1 , then D_1 is **NP-complete**.* ■

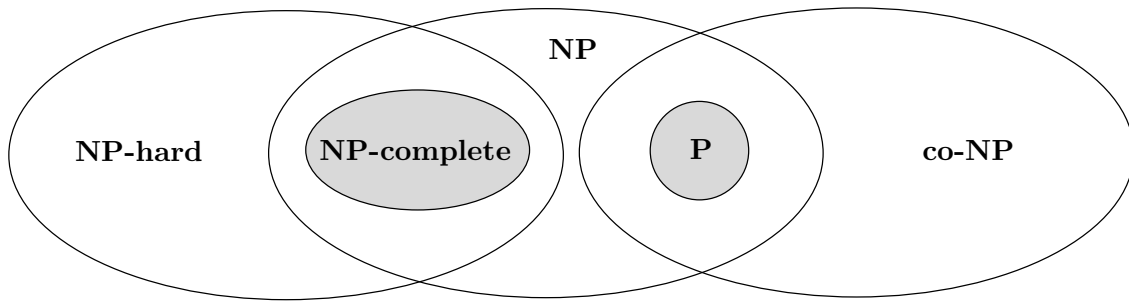


FIGURE 2.20: The complexity classes **P**, **NP**, **co-NP**, **NP-hard** and **NP-complete**.

2.3.3 The satisfiability decision problem

A *boolean variable* is a variable that can assume one of two values, **true** or **false**. The *negation* of a boolean variable x , denoted by \bar{x} , is another boolean variable which assumes the value **false** if and only if x assumes the value **true**. Let x_1, \dots, x_r be r boolean variables. Then a *literal* is either one of the variables x_i or its negation \bar{x}_i from the set of r boolean variables, while an *s-clause* is a conjunction of s literals, formed from the same set, conjoined by the binary operator **or**, denoted by \vee . A *boolean function* of the variables x_1, \dots, x_r is a function which maps the cartesian product

$$\underbrace{\{\mathbf{true}, \mathbf{false}\} \times \{\mathbf{true}, \mathbf{false}\} \times \cdots \times \{\mathbf{true}, \mathbf{false}\}}_{r \text{ sets}}$$

to the set $\{\mathbf{true}, \mathbf{false}\}$, and is said to be *satisfiable* if there exists an assignment of values from the set $\{\mathbf{true}, \mathbf{false}\}$ to the variables x_1, \dots, x_r for which the function evaluates to **true**. The *truth table* of a boolean function is a table in which the value of the function is listed against all of its possible variable values. Finally, a boolean function is in *s-conjunctive normal form* if the function comprises a number of s -clauses conjoined by the binary operation **and**, denoted by \wedge .

Consider, as an example, the boolean function

$$\phi^*(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee x_2 \vee x_4). \quad (2.3)$$

of four variables, which is in 3-conjunctive normal form. This function is satisfiable, as may be seen from its truth table (see Table 2.1).

x_1	x_2	x_3	x_4	$x_1 \vee x_2 \vee \overline{x_3}$	$x_2 \vee x_3 \vee \overline{x_4}$	$\overline{x_1} \vee x_2 \vee x_4$	ϕ^*
false	false	false	false	true	true	true	true
true	false	false	false	true	true	false	false
false	true	false	false	true	true	true	true
false	false	true	false	false	true	true	false
false	false	false	true	true	false	true	false
true	true	false	false	true	true	true	true
true	false	true	false	true	true	false	false
true	false	false	true	true	false	true	false
false	true	true	false	true	true	true	true
false	true	false	true	true	true	true	true
false	false	true	true	false	true	true	false
true	true	true	false	true	true	true	true
true	true	false	true	true	true	true	true
true	false	true	true	true	true	true	true
false	true	true	true	true	true	true	true
true	true	true	true	true	true	true	true

TABLE 2.1: Truth table for the boolean function $\phi^*(x_1, x_2, x_3, x_4)$ in (2.3).

The following famous decision problem forms one of the foundations of modern complexity theory.

Decision Problem 2.3 (*s*-SAT)

Instance: A boolean function $\phi(x_1, \dots, x_r)$ in *s*-conjunctive normal form.

Question: Is ϕ satisfiable?

The above decision problem is clearly in the class **NP**, a certificate being a set of boolean values for the variables x_1, \dots, x_r for which the function ϕ indeed evaluates to true. Cook [36] proved in a seminal paper in 1971 that Decision Problem 2.3 is, in fact, **NP-complete** for $s = 3$, which practically means that there is essentially no better method of solving the problem than considering all 2^r combinations of boolean values for the variables x_1, \dots, x_r in turn, until a combination is found for which the function ϕ evaluates to **true**. It is interesting to note, however, that Decision Problem 2.3 is in the class **P** for $s = 2$ [105, p. 500].

2.3.4 From decision problems to computation problems

A computational problem is one that has a real number (or a collection of real numbers) as solution instead of a binary variable. Computation problems may be solved by repeatedly solving their associated decision problems. Consider, as an example, the following decision problem.

Decision Problem 2.4 (Dominating set)

Instance: A graph $G = (V, E)$ and a positive integer $k \leq |V(G)|$.

Question: Does G have a dominating set of cardinality k or smaller?

By definition, it follows that $S = V(G)$ is a dominating set of a graph G of order n . Decision Problem 2.4 may therefore be solved repeatedly, each time decreasing the integer value k , starting

with $k = n - 1$, until no dominating set of cardinality $k = \ell < n$ can be found for the first time, in which case it follows that $\gamma(G) = \ell + 1$.

2.3.5 Graph domination is NP-complete

The following result dates back to 1979.

Theorem 2.36 (Garey & Johnson [49]) *Decision Problem 2.4 is NP-complete.* ■

This section is devoted to a description of the proof of Theorem 2.36. Let $C_{DS}(G)$ denote the computation problem of finding the domination number $\gamma(G)$ of a given graph G . Clearly Decision Problem 2.4 is in **NP**, a certificate to a problem instance being a dominating set of cardinality k for that instance. To prove that Decision Problem 2.4 is **NP-complete** it suffices to find a polynomial-time reduction of an instance of Decision Problem 2.3 (s -SAT) with $s = 3$ to an instance of Decision Problem 2.4 (Dominating set). The boolean function ϕ relevant to 3-SAT comprises k clauses in 3-conjunctive normal form.

Given an instance $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_k$ of 3-SAT, a corresponding instance G_ϕ of the domination problem may be constructed by mapping the function ϕ onto G_ϕ . A vertex C_i is created in G_ϕ to represent the corresponding clause in ϕ . For each decision variable x_j in ϕ a triangle is created in G_ϕ with vertices labelled x_j, \bar{x}_j and y_j . For each clause $C_i = x_j \vee x_\ell \vee x_m$ in G_ϕ the edges $(x_j, C_i), (x_\ell, C_i)$ and (x_m, C_i) are included in G_ϕ . It must be shown that ϕ has an assignment evaluating to **true** if and only if the graph G_ϕ contains a dominating set of cardinality k or smaller.

The graph G_{ϕ^*} constructed from the boolean function ϕ^* in (2.3), is shown as an example in Figure 2.21 for the boolean variable assignment $x_1 = x_4 = \mathbf{false}$ and $x_2 = x_3 = \mathbf{true}$. The graph G_{ϕ^*} in Figure 2.21 admits the dominating set $S = \{\bar{x}_1, x_2, x_3, \bar{x}_4\}$, and ϕ^* evaluates to **true** for the corresponding truth assignments of the boolean variables \bar{x}_1, x_2, x_3 and \bar{x}_4 , as indicated in Table 2.1.

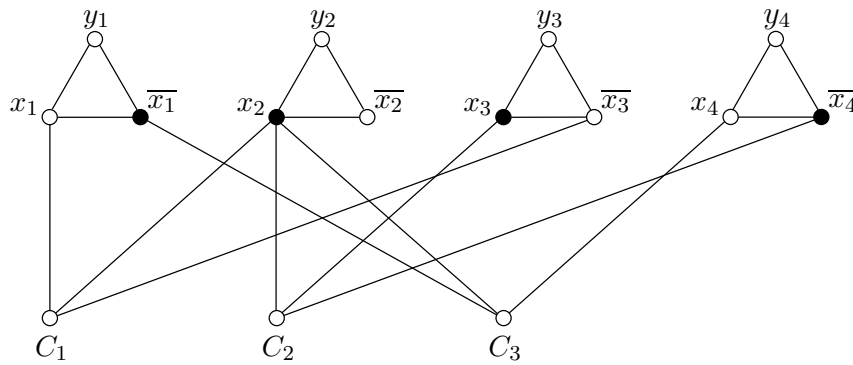


FIGURE 2.21: Reduction from the instance (2.3) of 3-SAT to an instance G_ϕ of the dominating set problem.

Suppose that ϕ admits a truth assignment and construct a subset D of the vertex set of G_ϕ by including vertex x_i in D if $x_i = \mathbf{true}$ or including \bar{x}_i in D if $x_i = \mathbf{false}$ in this assignment. Then the set D is a dominating set of G_ϕ , since each triangle of G_ϕ contains exactly one vertex of D and by assumption each vertex C_i is dominated by at least one vertex in D .

Conversely, suppose that G_ϕ has a dominating set D of cardinality k . Each of the vertices y_1, \dots, y_r must either be in D or dominated by a vertex in D . Therefore, each triangle of G_ϕ

must contain exactly one vertex in D . But since $|D| = k$ and G_ϕ has k triangles, D contains none of the clause vertices C_1, \dots, C_k . However, since D is a dominating set, each clause vertex must be dominated by at least one vertex in D . The function ϕ is therefore satisfied by the boolean assignments

$$x_i = \begin{cases} \text{true} & \text{if } x_i \in D \\ \text{false} & \text{if } \bar{x}_i \in D \end{cases}$$

for all $i = 1, \dots, r$.

The length of the instance ϕ of 3-SAT is $3r + k$, since the function ϕ consists of r literals and k clauses, and each clause contains three decision variables. The graph G_ϕ consists of $3r + k$ vertices and $3r + 3k$ edges. Therefore, the cardinality of G_ϕ is at most a constant times the number of literals in ϕ , and thus the graph G_ϕ can be constructed from an instance of 3-SAT in polynomial-time.

2.3.6 Algorithms for computing the domination number of a graph

Given a multiset \mathcal{S} of sets over a universe \mathcal{U} , with the property that $\bigcup_{S \in \mathcal{S}} S = \mathcal{U}$, a *set cover* \mathcal{V} of \mathcal{S} is a subset $\mathcal{V} \subseteq \mathcal{S}$ with the property that

$$\bigcup_{S \in \mathcal{V}} S = \mathcal{U}.$$

A set cover \mathcal{V} is called a *minimum set cover* if it is of minimum cardinality. Consider the following decision problem.

Decision Problem 2.5 (Set cover)

Instance: A multiset of sets \mathcal{S} over a universe \mathcal{U} and a positive integer $k \leq |\mathcal{S}|$.

Question: Does there exist a set cover $\mathcal{V} \subseteq \mathcal{S}$ of \mathcal{U} of cardinality k or smaller?

Decision Problem 2.4 is clearly the special case of Decision Problem 2.5 in which the universe \mathcal{U} is taken as the vertex set of the graph G in Decision Problem 2.4 and in which the multiset \mathcal{S} is taken as the set of closed neighbourhoods of the vertices of G in Decision Problem 2.4.

In the tree T_1 in Figure 2.22(a), for example, the universe is $\mathcal{U} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, while the multiset \mathcal{S} is $\{\{v_1, v_2, v_3\}, \{v_1, v_2\}, \{v_1, v_3, v_4\}, \{v_3, v_4, v_5, v_6\}, \{v_4, v_5\}, \{v_4, v_6\}\}$. Clearly, the union of all the sets in \mathcal{S} contain all elements in \mathcal{U} . However, an example of a minimum set cover of \mathcal{U} is $\{\{v_1, v_2, v_3\}, \{v_3, v_4, v_5, v_6\}\}$. Since T_1 contains no universal vertex, it follows that $\gamma(T_1) = 2$, with a minimum dominating set of T_1 being $S = \{v_1, v_4\}$, the set of vertices whose closed neighbourhoods form the minimum set cover, as shown in Figure 2.22(b).



FIGURE 2.22: (a) A tree T_1 and (b) a minimum dominating set of T_1 .

A trivial recursive algorithm for computing minimum set covers is shown in Algorithm 2.2, which takes a multiset \mathcal{S} of a universe \mathcal{U} as input. If \mathcal{S} is empty and \mathcal{U} is not empty, the algorithm terminates that branch of the search tree in Step 2, as it will not yield a cover of \mathcal{U} . If both \mathcal{S} and \mathcal{U} are empty, the algorithm returns an empty set in Step 5. This occurs when a branch of the search tree has found a cover of \mathcal{U} . In the case \mathcal{S} is not empty, the algorithm selects an element $S \in \mathcal{S}$ of maximum cardinality in Step 6 and a branching decision is taken with respect to S , namely either to include S in a potential cover of \mathcal{U} , or to exclude S from the cover of \mathcal{U} :

- The inclusion of S in the cover over \mathcal{U} reduces the universe to $\mathcal{U} - S$ and each set $S' \in \mathcal{S} - S$ is reduced to $S' - S$.
- In the latter case the set S is removed from the multiset \mathcal{S} .

The algorithm recursively solves both subproblems and returns the smallest cover found by the recursive calls.

Algorithm 2.2: MSC

Input : A set cover instance $(\mathcal{S}, \mathcal{U})$.

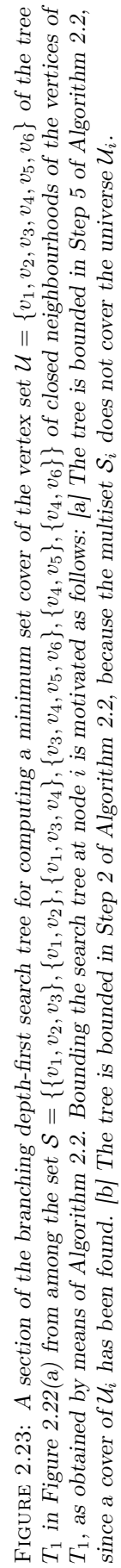
Output: A minimum set cover of \mathcal{U} from the elements of \mathcal{S} .

```

1 if  $\mathcal{S} = \emptyset$  and  $\mathcal{U} \neq \emptyset$  then
2   return [False];
3 else
4   if  $\mathcal{S} = \emptyset$  then
5     return  $[\emptyset]$ ;
6   Let  $S \in \mathcal{S}$  be a set of maximum cardinality;
7   return  $[\min \{ \{S\} \cup \text{MSC}(\{S' - S \mid S' \in \mathcal{S} - \{S\}\}, \mathcal{U} - S), \text{MSC}(\mathcal{S} - \{S\}, \mathcal{U}) \}]$ ;
```

The working of Algorithm 2.2 is illustrated in Figure 2.23 which contains a part of the search tree constructed by the algorithm when computing a minimum set cover of $\mathcal{U} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ from the multiset $\mathcal{S} = \{\{v_1, v_2, v_3\}, \{v_1, v_2\}, \{v_1, v_3, v_4\}, \{v_3, v_4, v_5, v_6\}, \{v_4, v_5\}, \{v_4, v_6\}\}$ corresponding to the associated dominating set problem for the tree T_1 in Figure 2.22(a). The set $\{v_3, v_4, v_5, v_6\} \in \mathcal{S}$ has the largest cardinality and therefore the algorithm branches on this set, producing the two subproblems shown in nodes 1 and 8 of the search tree in Figure 2.23. The tree is traversed in a depth-first fashion. The problem at node 1 consists of the set $\{v_3, v_4, v_5, v_6\}$ together with the output of Algorithm 2.2 when called with the multiset $\mathcal{S}_1 = \{\{v_1, v_2\}, \{v_1, v_2\}, \{v_1\}\}$ and the universe $\mathcal{U}_1 = \{v_1, v_2\}$ as input. The set $\{v_1, v_2\} \in \mathcal{S}$ is chosen next to branch upon. This time a cover of the universe is found, since $\mathcal{S}_2 = \emptyset$ and $\mathcal{U}_2 = \emptyset$. The search tree is therefore bounded in Step 5 of Algorithm 2.2, denoted by bounding reason “[a]” in the figure. Backtracking to node 3, the set $\{v_1, v_2\}$ is excluded from the cover such that $\mathcal{S}_3 = \{\{v_1, v_2\}, \{v_1\}\}$ and the set $\mathcal{U}_3 = \{v_1, v_2\}$ is next branched upon, leading to nodes 4 and 5 of the search tree. Node 4 produces a cover of \mathcal{U}_4 . At node 5 the set $\{v_1, v_2\}$ is excluded from the cover such that $\mathcal{S}_5 = \{\{v_1\}\}$. The set $\{v_1\}$ is then branched upon, leading to nodes 6 and 7 of the search tree. The multisets \mathcal{S}_6 and \mathcal{S}_7 in nodes 6 and 7 cannot cover the universes \mathcal{U}_6 and \mathcal{U}_7 , respectively, and so the search tree is bounded in Step 2 of Algorithm 2.2, denoted by bounding reason “[b]” in the figure. Thereafter the search backtracks to node 8 and the same procedure may be followed in the remaining nodes of the search tree. The worst-case time complexity of Algorithm 2.2 is $\mathcal{O}(2^n)$ for a graph of order n .

The earliest known algorithm that improved upon the time complexity of the trivial approach of Algorithm 2.2 for finding a minimum dominating set of an arbitrary graph is due to Fomin *et*



al. [46] and dates to 2004. The algorithm of Fomin *et al.* [46] uses the result of Theorem 2.10, which provides an upper bound on the domination number of graphs of minimum degree at least three. An initial polynomial-time algorithm, based on a pruning search tree technique, eliminates all vertices of degree one or two from the input graph G . The algorithm terminates with a graph G' containing only vertices of degree zero or at least three in G . Let V' be the set of vertices of degree at least three in G' , let $n = |V'|$ and let $G'' = \langle V' \rangle_G$. Theorem 2.10 guarantees the existence of a dominating set of G'' of cardinality at most $3n/8$. Fomin *et al.* [46] then test all possible subsets of $V(G'')$ with up to $3n/8$ vertices to find a minimum dominating set of G'' . By using Sterling's approximation $x! \approx x^x e^{-x} \sqrt{2\pi x}$ for factorials, and by suppressing some polynomial factors, the number of subsets that must be tested for finding a minimum dominating set X of G'' is at most

$$\binom{n}{3n/8} = \frac{(n)!}{(3n/8)! (5n/8)!} = \mathcal{O}(8^n \cdot 3^{-3n/8} \cdot 5^{-5n/8}) = \mathcal{O}(1.9379^n). \quad (2.4)$$

The vertices of G of degree zero are then added to X in polynomial-time to obtain a minimum dominating set of G .

Grandoni [54] designed an algorithm for the set cover problem, where k is the sum of the number of sets in the multiset \mathcal{S} and the cardinality of \mathcal{U} . The dominating set problem is formulated as a minimum set cover problem of dimension $k = 2n$. His polynomial space recursive algorithm for finding a minimum dominating set runs in

$$\mathcal{O}(1.3803^{2n}) = \mathcal{O}(1.9053^n) \text{ time.} \quad (2.5)$$

The approach of Grandoni incorporates two reduction rules into the trivial set cover approach of Algorithm 2.2 in order to reduce the size of search tree. The two reduction rules are based on the following observations:

- If there are sets $S, R \in \mathcal{S}$, such that $S \subset R$, then there is a minimum set cover of \mathcal{U} which does not contain S .
- If there is an element u which belongs to a unique set $S \in \mathcal{S}$, then S belongs to every set cover over \mathcal{U} .

Grandoni improved on the complexity in (2.5) by employing a dynamic programming approach to reduce the time complexity of algorithm, storing all solutions of subproblems in a database. When branching the search tree of a set cover problem, subproblems may be identical to those solved at an earlier stage in the search tree. The database is therefore queried to determine whether a solution to a subproblem is already available. This version of the algorithm for finding a minimum dominating set can be solved in

$$\mathcal{O}(1.3424^{2n}) = \mathcal{O}(1.8021^n) \text{ time at the expense of an exponential space complexity.} \quad (2.6)$$

Schiermeyer [88] designed a polynomial space algorithm for finding a minimum dominating set of an arbitrary graph G in $\mathcal{O}(1.8899^n)$ time. An approach similar to that of Fomin *et al.* [46] was used to eliminate a subset of vertices of G in order to obtain a graph G' for which $\gamma(G') \leq \frac{n}{3}$. The algorithm for finding a minimum dominating set of G' commences by partitioning the vertices of G' into three distinct subsets. Schiermeyer used a series of matching techniques to determine a minimum dominating set X of G' and he showed that there exists a subset of vertices $X' \subseteq V(G) - V(G')$ such that $X \cup X'$ is a minimum dominating set of G .

In 2005, Fomin *et al.* [47] improved on the time complexity in (2.6) by incorporating the following additional reduction rule

- If $S \in \mathcal{S}$ is an element of maximum cardinality and $|S| \leq 2$, then a minimum set cover of \mathcal{U} may be computed in polynomial-time by finding a maximum matching [40]. A maximum matching is found for an input graph G , where each element in \mathcal{U} is a vertex of G , and for each set $S' = \{e_1, e_2\}$ in \mathcal{S} such that $|S'| = 2$, e_1e_2 is an edge of G .

Van Rooij, often together with others, has in recent years made significant contributions towards good algorithms for graph domination by adopting a measure and conquer approach. Van Rooij *et al.* [98], for example, improved on the algorithm by Fomin *et al.* [47] by designing an exponential space algorithm for finding all dominating sets of cardinality k of a graph G of order $n \geq k$ in $\mathcal{O}(1.5048^n)$ time, also using a search tree and making use of inclusion/exclusion branching rules.

Van Rooij and Bodlaender's [97] most recent polynomial-time algorithm of 2011 is based on the minimum set cover problem, using a series of reduction rules to reduce the running time of the algorithm. In addition to the reduction rules presented by Grandoni [54] and Fomin *et al.* [47], they were able to include four more reduction rules.

Table 2.2 contains a summary of the time complexities of the algorithms for computing the domination number of a graph discussed so far in this section.

Authors	Year	Polynomial-space	Exponential-space
Fomin <i>et al.</i> [46]	2004	$\mathcal{O}(1.9379^n)$	
Grandoni [54]	2006	$\mathcal{O}(1.9053^n)$	$\mathcal{O}(1.8021^n)$
Schiermeyer [88]	2008	$\mathcal{O}(1.8899^n)$	
Fomin <i>et al.</i> [47]	2005	$\mathcal{O}(1.5263^n)$	$\mathcal{O}(1.5086^n)$
Van Rooij <i>et al.</i> [98]	2009		$\mathcal{O}(1.5048^n)$
Van Rooij & Bodlaender [97]	2011	$\mathcal{O}(1.4969^n)$	

TABLE 2.2: Worst-case time complexities of exact algorithms for computing the domination number of an arbitrary graph of order n .

Cockayne, Goodman and Hedetniemi [31] designed the first linear algorithm for determining the domination number of a tree in 1975. In order to describe their algorithm, the notion of a canonical ordering is required. An *ordering* of the vertices of a tree T of order n is an assignment of the indices $1, \dots, n$ to the vertices of T , one index to a vertex. A *canonical ordering* of the vertices of a rooted tree T is an ordering of the vertices of T such that the index of the parent of vertex i , denoted by $\text{Parent}[i]$, is smaller than i . The root of T therefore has index 1 and the special convention is adopted where the “index” of $\text{Parent}[1]$ is 0.

Cockayne *et al.* [31] partitioned the vertex set of an arbitrary tree into three subsets, a set V_1 of **Required** vertices, a set V_2 of **Bound** vertices and a set V_3 of **Free** vertices. An *optional dominating set* of a tree is a set of vertices D which contains all **Required** vertices (*i.e.* $V_3 \subseteq D$) and dominates all **Bound** vertices. **Free** vertices need not be dominated by D but may be included in D in order to dominate **Bound** vertices. The *optional dominating number* of a tree T is the minimum cardinality of an optional dominating set and is denoted by $\gamma_{\text{opt}}(T)$. For an arbitrary tree T with vertex set $V(T) = V_1 \cup V_2 \cup V_3$ as described above, the algorithm commences by setting $V(T) = V_2$ (*i.e.* $V_1 = V_3 = \emptyset$, in which case clearly $\gamma_{\text{opt}}(T) = \gamma(T)$). As the algorithm progresses, vertices are added to the set V_1 depending on their indices in the canonical ordering of $V(T)$. When a vertex i is encountered, its index $\text{Label}[i]$ together with that of its parent, $\text{Parent}[i]$, are used to possibly relabel $\text{Parent}[i]$ to either **Free** or **Required**. Once a vertex has

Algorithm 2.3: Tree domination

Input : A tree T represented by an array $\text{Parent}[1, \dots, n]$.
Output: A minimum dominating set of T , represented by the set V_1 .

```

1  $V_1 \leftarrow \emptyset$ ;
2 for  $i = 1$  to  $n$  do
3    $\text{Label}[i] \leftarrow \text{Bound}$ ;
4 for  $i \leftarrow n$  down to 2 do
5   if  $\text{Label}[i] = \text{Bound}$  then
6      $\text{Label}[\text{Parent}[i]] \leftarrow \text{Required}$ ;
7   else if  $\text{Label}[i] = \text{Required}$  then
8      $V_1 \leftarrow V_1 \cup \{i\}$ ;
9     if  $\text{Label}[\text{Parent}[i]] = \text{Bound}$  then
10       $\text{Label}[\text{Parent}[i]] \leftarrow \text{Free}$ ;
11 if ( $\text{Label}[1] = \text{Bound}$ ) or ( $\text{Label}[1] = \text{Required}$ ) then
12    $V_1 \leftarrow V_1 \cup \{1\}$ ;
13 return  $V_1$ ;

```

been labelled **Required** its label does not change again. Upon completion of the algorithm, all the vertices labelled **Required** form a minimum dominating set of T .

Figure 2.24 shows the rooted tree of Figure 2.5 together with its parent array. The solid vertices denote the **Required** vertices as determined by Algorithm 2.3.

The following result guarantees that Algorithm 2.3 will find a minimum cardinality dominating set for any tree.

Theorem 2.37 (Cockayne *et al.* [31]) *If T is a tree, then the set of vertices V_1 designated as **Required** by Algorithm 2.3 is a minimum dominating set of T .* ■

It is easy to see that Algorithm 2.3 runs in $\mathcal{O}(n)$ time, as a **for**-loop is merely executed, once from $i = 1$ to $i = n$, and all of the statements within the **for**-loop can be performed in constant time.

2.4 Chapter summary

In this chapter, the basic terminology and most important results from graph theory that are applicable to this dissertation, were introduced. The most basic fundamentals from graph theory were reviewed in §2.1, after which the focus shifted to operations on graphs and common characteristics of graphs in general. An overview of well-known special classes of graphs was also included.

In §2.2, important properties relating to graph domination were discussed. Maximality, minimality and (super)hereditary properties were used as a basis for reviewing the close relationships between domination, independence and irredundance, resulting in the inequality chain involving the domination parameters, as described in Theorem 2.5. Other fundamental contributions to graph domination, notably the work of Berge [6] and Ore [80], as well as various bounds on the

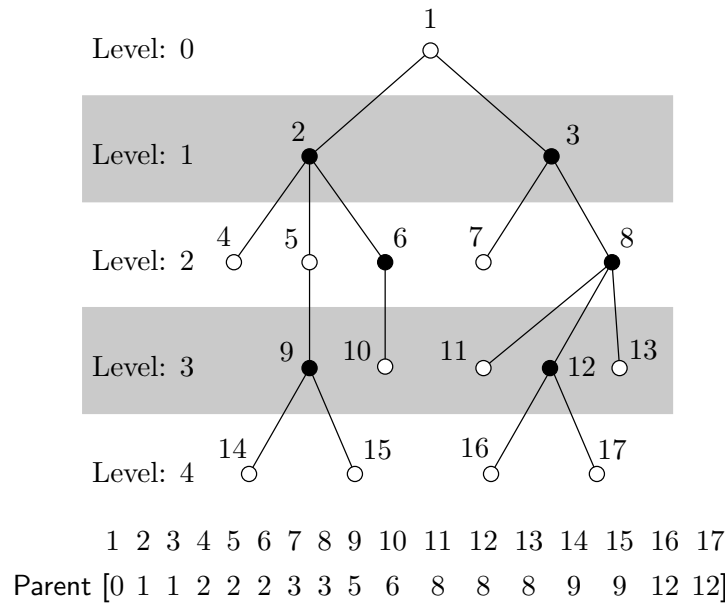


FIGURE 2.24: A rooted tree T_2 together with its parent array. The solid vertices form a minimum dominating set of T , found by Algorithm 2.3.

domination number of a graph, were also highlighted. An overview of the literature on the effects of edge deletion and addition on graph domination was also presented.

The focus turned to basic complexity theoretic concepts in §2.3. More specifically, the notions of time and space complexities of algorithms were discussed, and this was followed by an overview of the complexity classes of decision problems **P** and **NP**. The notion of polynomial-time reducibility was demonstrated in the context of the classes **P** and **NP**, and it was shown that the dominating set decision problem is **NP-complete**. Recent algorithms for determining the domination number of a graph were also reviewed, adopting as a point of departure the trivial set cover algorithm (Algorithm 2.2), which laid the foundation for Van Rooij and Bodlaender's 2011 algorithm for finding the dominating number of an arbitrary graph in $\mathcal{O}(1.4969^n)$ time. Cockayne, Goodman and Hedetniemi's 1975 linear algorithm for finding the domination number of a tree was also reviewed and illustrated.

CHAPTER 3

Literature review

Contents

3.1	Basic notions of graph protection	47
3.2	Secure graph domination	52
3.2.1	Bounds on γ_s	52
3.2.2	Special graph classes	54
3.2.3	Edge removal criticality in secure graph domination	56
3.3	Generalised graph protection parameters	59
3.3.1	Finite higher order generalisations	59
3.3.2	Generalisations to include infinitely many moves	61
3.3.3	Generalisations involving multiple moves	61
3.4	Chapter summary	62

This chapter opens with an overview of some well-known graph protection parameters, including those associated with graph domination, Roman domination, total domination, weak Roman domination and secure domination. A survey of results from the literature on secure graph domination is presented in §3.2. A natural generalisation of the classical parameters related to graph protection is finally reviewed in §3.3.

3.1 Basic notions of graph protection

During the fourth century A.D., the Roman Empire had a total of twenty five legions at its disposal to defend its territories. Each legion consisted of various infantry and cavalry units [74]. A grouping of six legions, called a *field army*, was deemed sufficient to secure any one of the eight regions represented by the vertices of the graph Ξ superimposed on the map of the empire in Figure 3.1. Emperor Constantine the Great (274–337 A.D.) therefore commanded four field armies and had to decide how to deploy these field armies. The Emperor considered a deployment capable of securing the entire empire if every one of its eight regions was either occupied by a field army or was directly adjacent to a region occupied by two field armies [85], where adjacencies of these regions are indicated by the edges of the graph Ξ in Figure 3.1 (these edges represented deployment routes at the time). Constantine’s reasoning was that two field armies had to be stationed in a region before one would be allowed to move to a neighbouring, unoccupied region in order to deal with an internal uprising or external defence challenge there, so that one of the

field armies could help launch the other and so that the region vacated by the moving field army could not immediately be attacked successfully by an enemy.

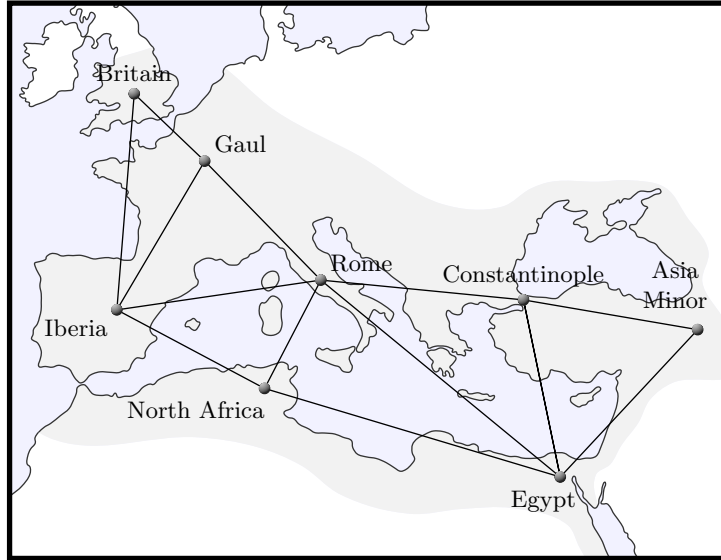


FIGURE 3.1: A graph model Ξ of the eight regions of the Roman Empire during the fourth century A.D.

It is not immediately obvious whether or not the entire empire could have been defended by only four field armies. Emperor Constantine, in fact, chose to sacrifice Britain by securing the central regions of the Empire when he stationed two field armies in Rome and two in Constantinople, as shown in Figure 3.2(a). He could, however, have secured the entire empire by rather stationing one field army in Britain, two in Rome and one in Asia Minor, as shown in Figure 3.2(b). The defence strategy of Emperor Constantine is one of the earliest recorded facility location problems and is, in fact, a generalisation of the modern notion of *domination* in graph theory, formalised in the seminal work of Berge [6] and Ore [80] during the period 1958–1962. The subfield of graph domination has numerous applications other than the safeguarding of territories, but in keeping with the application described above, the more generic term *guard* is used instead of *field army* to explain the concept of graph domination and its variations in the remainder of this section.

As described in Chapter 2, a *dominating set* of a graph G is a subset X of the vertex set of G , where X represents those vertices of G that receive one guard each, with the property that each vertex of G which is not in X should be adjacent to at least one vertex in X . Recall that the *domination number* of G , denoted by $\gamma(G)$, is the minimum value of $|X|$, taken over all dominating sets X of G (i.e. the smallest number of guards that can possibly form a dominating set of G). The notion of domination may also be defined in terms of guard functions.

Let $f : V(G) \mapsto \{0, 1, 2, \dots\}$ be a function, called a *guard function* of G , where $f(v)$ is the number of guards placed at $v \in V(G)$. Let V_i be the subset of vertices in $V(G)$ for which $f(v) = i$, for all $i = 0, 1, 2, \dots$. A guard function f may therefore be specified by the partition of the vertex set of G as $f = (V_0, V_1, V_2, \dots)$. A guard function is *safe* if each vertex $v \in V_0$ is adjacent to at least one vertex in $V - V_0$, or equivalently, if $V(G) - V_0$ is a dominating set of G . For any safe guard function $f = (V_0, V_1, V_2, \dots)$, the *weight* of f is defined as $w(f) = \sum_{v \in V(G)} f(v) = \sum_{i \geq 1} i|V_i|$, while $f(S) = \sum_{v \in S} f(v)$ for any subset $S \subseteq V(G)$. Note, therefore, that $w(f) = f(V(G))$ for any graph G . In this chapter, the notation $f(N[v])$ is abbreviated to $f[v]$ for convenience.

A dominating set V_1 of a graph G is therefore a set for which the guard function $f = (V_0, V_1)$ is a safe guard function of G , and the domination number $\gamma(G)$ of G is the smallest weight that a safe guard function of G can assume. A dominating set of smallest cardinality for the graph Ξ

in Figure 3.1 is depicted in Figure 3.2(c), showing that $\gamma(\Xi) = 2$. The notion of domination in graphs has attracted significant attention in the graph theory literature since the 1970s [59, 60] and is a special case of the celebrated *set cover problem* in the operations research literature, as described in the work of Caprara *et al.* [24].

A very natural variation on the theme of domination is that of *total domination*, introduced by Cockayne *et al.* [28] in 1980. A *total dominating set* of a graph G is a subset X_t of the vertex set of G , in which each vertex receives one guard, with the property that *every* vertex of G should be adjacent to at least one vertex in X_t (*i.e.* in addition to the set being a dominating set, every vertex in X_t should also be adjacent to at least one other vertex in X_t). The *total domination number* of G , denoted by $\gamma_t(G)$, is the minimum value of $|X_t|$, taken over all total dominating sets X_t of G (*i.e.* the smallest number of guards that can possibly form a total dominating set of G). The guard function $f = (V_0, V_1)$ associated with a total dominating set V_1 of G , where $V_0 = V(G) - V_1$, is therefore a safe guard function of G with the additional property that

$$\sum_{v \in N(u)} f(v) \geq 1 \quad \text{for all } u \in V_1.$$

A total dominating set of smallest cardinality for the graph Ξ in Figure 3.1 is shown in Figure 3.2(d), from which it follows that $\gamma_t(\Xi) = 3$. It is immediately obvious that $\gamma_t(G) \geq \gamma(G)$ for any graph G . The notion of total domination was inspired by policing and monitoring applications in which the total dominating set represents vertices at which guards are placed, but with the additional requirement that each guard should itself also be monitored by at least one other guard for auditing purposes in an attempt at safeguarding against corruption of the guards. The notion of total domination has been studied by various authors [28, 41, 61, 64, 65, 73].

Bollobás and Cockayne [7] proved the following result in 1979.

Theorem 3.1 (Bollobás & Cockayne [7]) *For any graph G without isolated vertices, there exists a minimum dominating set X of the vertex set of G for which every vertex $v \in X$ has the property that $|\text{Epn}(v, X)| \geq 1$.* ■

Henning [61], who perhaps was not the first, established the following relationship between the domination number and the total domination number of a graph without any isolated vertices. This result follows trivially from Theorem 3.1.

Proposition 3.1 (Henning [61]) *For any graph G without isolated vertices,*

$$\gamma(G) \leq \gamma_t(G) \leq 2\gamma(G). \quad \blacksquare$$

Constantine's defence strategy of the Roman Empire represents another variation on the concept of domination. A *Roman dominating function* of a graph G is a safe guard function $f = (V_0, V_1, V_2)$ of G , with the additional property that each vertex in V_0 should be adjacent to at least one vertex in V_2 . The *Roman domination number* of G , denoted by $\gamma_R(G)$, is the minimum value of $|V_1| + 2|V_2|$, taken over all Roman dominating functions $f = (V_0, V_1, V_2)$ of G (*i.e.* the smallest weight that a Roman dominating function of G can assume). It is not difficult to show that $\gamma_R(\Xi) = 4$, *i.e.* that the Roman defence strategy in Figure 3.2(b) is best possible. The notion of Roman domination in graphs has been studied by various authors [29, 32, 87, 91].

Cockayne *et al.* [29] established the following bounds on the Roman domination number of a graph in 2004.

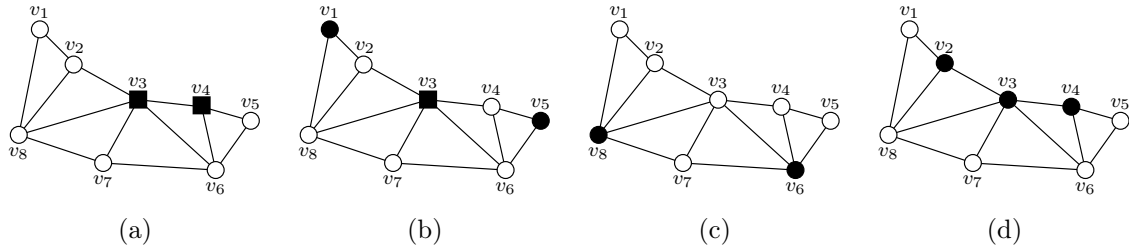


FIGURE 3.2: (a) Emperor Constantine's defence strategy was to station two field armies in Rome and two in Constantinople. (b) A better defence strategy, avoiding the sacrifice of Britain. (c) A dominating set of minimum cardinality for the graph Ξ . (d) Both a weak Roman and secure dominating set of minimum cardinality for the graph Ξ . In all cases v_1 denotes Britain, v_2 denotes Gaul, v_3 denotes Rome, v_4 denotes Constantinople, v_5 denotes Asia Minor, v_6 denotes Egypt, v_7 denotes North Africa, v_8 denotes Iberia. Furthermore, \circ represents an unoccupied vertex, \bullet represents a vertex occupied by one guard or field army and \blacksquare represents a vertex occupied by two guards or field armies.

Theorem 3.2 (Cockayne *et al.* [29]) *For any graph G ,*

$$\gamma(G) \leq \gamma_R(G) \leq 2\gamma(G).$$

Proof: Let $f = (V_0, V_1, V_2)$ be a Roman dominating function of G and let X be a dominating set of G . It follows that $V_1 \cup V_2$ is a dominating set of G and that $(V(G) - X, \emptyset, X)$ is a Roman dominating function of G . Hence, $\gamma(G) \leq |V_1| + |V_2| \leq |V_1| + 2|V_2| = \gamma_R(G)$. Moreover, $\gamma_R(G) \leq 2|X| = 2\gamma(G)$. ■

Under the assumption that no two regions of the Roman Empire would be attacked simultaneously, Emperor Constantine could have defended the empire using even fewer than four of his thinly stretched field armies. This observation led Henning and Hedetniemi [62] to introduce the notion of weak Roman domination in 2003. A *weak Roman dominating function* of a graph G is a safe guard function $f = (V_0, V_1, V_2)$ of G , with the additional property that, for each vertex $u \in V_0$, there exists a vertex $v \in V_1$ for which the *swap set* $(V_1 \cup \{u\}) - \{v\} \cup V_2$ is again a dominating set of G , or a vertex $v \in V_2$ for which the *swap set* $(V_2 - \{v\}) \cup (V_1 \cup \{u, v\})$ is again a dominating set of G . The notion of a swap set models the situation where a guard moves from a single occupied vertex v or a doubly occupied vertex v' to an unoccupied vertex u in order to deal with a problem at u , but leaving the resulting configuration a safe guard function of G again. The vertex v (or v' in the latter case) is said to *defend* u . The *weak Roman domination number* of G , denoted by $\gamma_r(G)$, is the minimum value of $|V_1| + 2|V_2|$, taken over all weak Roman dominating functions (V_0, V_1, V_2) of G (i.e. the smallest weight that a weak Roman dominating function can assume). The minimum total dominating set in Figure 3.2(d) is incidentally also a weak Roman dominating set of minimum cardinality for the graph Ξ in Figure 3.1. To see this, note that the guard at v_2 (field army in Gaul) is able to move to either of the vertices v_1 or v_8 (Britain or Iberia) if a security threat were to occur there. The guard at v_3 (field army in Rome) can similarly defend the unoccupied vertex v_7 (North Africa), while the guard at v_4 (field army in Constantinople) can defend the unoccupied vertices v_5 or v_6 (Asia Minor or Egypt). It is not too difficult to show that the weak Roman dominating set in Figure 3.2(d) is best possible and hence that $\gamma_r(\Xi) = 3$. Note that a total dominating set of a graph G is not always a weak Roman dominating set of G ; it is a mere coincidence for the graph Ξ .

Henning and Hedetniemi [62] were able to include the weak Roman domination of a graph into the inequality chain of Theorem 3.2.

Theorem 3.3 (Henning & Hedetniemi [62]) For any graph G ,

$$\gamma(G) \leq \gamma_r(G) \leq \gamma_R(G) \leq 2\gamma(G).$$

Proof: It is first shown that every Roman dominating function in a graph G is also a weak Roman dominating function of G . Let (V_0, V_1, V_2) be a Roman dominating function of G . Suppose $u \in V_0$. Then u is adjacent to a vertex $v \in V_2$. Let (V'_0, V'_1, V'_2) be the guard function such that $V'_2 = V_2 - \{v\}$ and $V'_1 = V_1 \cup \{u, v\}$ (i.e. (V'_0, V'_1, V'_2) corresponds to the swap set $(V_2 - \{v\}) \cup (V_1 \cup \{u, v\})$). Each vertex $w \in V'_0$ is adjacent to v or a vertex in V'_2 and is therefore dominated. Thus, (V'_0, V'_1, V'_2) is a safe guard function of G and is therefore a weak Roman dominating function of G , and so $\gamma_r(G) \leq \gamma_R(G)$. To show that $\gamma_r(G) \geq \gamma(G)$, suppose (V''_0, V''_1, V''_2) is a weak Roman dominating function of G of minimum weight. Since $V''_1 \cup V''_2$ is a dominating set of G , it follows that $\gamma(G) \leq |V''_1 \cup V''_2| = |V''_1| + |V''_2| \leq |V''_1| + 2|V''_2| = \gamma_r(G)$. ■

Whereas the possibility of placing two guards at a vertex within the context of Roman domination is historically well-founded, this seems to be an artificial construct in the relaxed setting of weak Roman domination. This observation led by Cockayne *et al.* [32] in 2004 to the simpler notion of *secure domination* where each vertex of the graph can accommodate at most one guard. A *secure dominating set* of a graph G is therefore a subset X_s of the vertex set of G , where X_s represents those vertices of G that receive one guard each, with the property that X_s forms a dominating set of G and additionally, for each vertex u not in X_s , there exists a vertex $v \in X_s$ such that the *swap set* $(X_s - \{v\}) \cup \{u\}$ is again a dominating set of G . Here the swap set again models the situation where a guard moves from an occupied vertex v to an unoccupied vertex u , again leaving the resulting configuration a dominating set and hence v defends u . The *secure domination number* of G , denoted by $\gamma_s(G)$, is the minimum value of $|X_s|$, taken over all secure dominating sets X_s of G (i.e. the smallest number of guards that can form a secure dominating set of G). The guard function $f = (V_0, V_1)$ associated with a secure dominating set X_s of G is therefore a safe guard function of G for which $V_1 = X_s$ and $V_0 = V(G) - X_s$ with the additional property that, for each vertex $u \in V_0$, there exists a vertex $v \in V_1$ such that $((V_0 - \{u\}) \cup \{v\}, (V_1 - \{v\}) \cup \{u\})$ is also a safe guard function of G . Since each vertex in the weak Roman dominating set of the graph Ξ shown in Figure 3.2(d) accommodates a single guard already, it is also a secure dominating set of G .

Cockayne *et al.* [32] were able to incorporate the secure domination number of a graph G into the inequality chain in Theorem 3.3.

Theorem 3.4 (Cockayne *et al.* [32]) For any connected graph G ,

$$\gamma(G) \leq \gamma_r(G) \leq \begin{cases} \gamma_R(G) \leq 2\gamma(G) \\ \gamma_s(G). \end{cases}$$

Proof: It is shown that every secure dominating function of G is also a weak Roman dominating function of G . Let (V_0, V_1) be a secure dominating function of G . Each vertex $u \in V_0$ is adjacent to some vertex $v \in V_1$. Let the guard function (V'_0, V'_1) correspond to the swap set $V'_1 = (V_1 - \{v\}) \cup \{u\}$ and $V'_0 = (V_0 - \{u\}) \cup \{v\}$. Each vertex $w \in V'_0$ is adjacent to some vertex in V'_1 and is therefore dominated. Thus, (V'_0, V'_1) is a safe guard function of G and (V_0, V_1, \emptyset) is therefore a weak Roman dominating function of G . It follows that $\gamma_r(G) \leq \gamma_s(G)$. ■

It is possible to show that $\gamma_s(\Xi) = 3$. Although $\gamma_r(\Xi) = \gamma_s(\Xi)$, the parameters $\gamma_r(G)$ and $\gamma_s(G)$ differ for graphs in general. Cockayne *et al.* [30] were able to establish the following result in 2003.

Theorem 3.5 (Cockayne *et al.* [30]) *If G is claw-free, then $\gamma_r(G) = \gamma_s(G)$.* ■

Benecke *et al.* [5] introduced the notion of *secure total domination* and were able to provide some basic properties of secure total dominating sets. A subset of vertices S of a graph G is a *secure total dominating set* of G if S is a total dominating set of G and there exists, for each vertex $u \in V(G) - S$, some vertex $v \in S \cap N(u)$ such that $(S - \{v\}) \cup \{u\}$ is a total dominating set of G .

The *secure total domination number* of a graph G is denoted by $\gamma_{st}(G)$. Benecke *et al.* [5] established properties of secure total dominating sets of graphs in general, determined $\gamma_{st}(\mathcal{P}_n)$ for all values of n , and established a lower bound on the secure total domination number of a forest with maximum degree larger than two. Klostermeyer and Mynhardt [68] presented results on the relationships between the secure domination numbers and the secure total domination numbers of graphs with specific properties. They showed, for example, that the secure total domination number of an isolate-free graph is at most twice its clique covering number¹, and no more than three times its independence number, and additionally, demonstrated that the first bound is sharp. Furthermore, $\gamma_s(G) \leq 2\beta(G)$ for any graph G [68].

3.2 Secure graph domination

An important characterisation of secure dominating sets, which is due to Cockayne *et al.* [32], dates back to 2005.

Theorem 3.6 (Cockayne *et al.* [32]) *Let X be a dominating set of G . Then a vertex $v \in X$ defends a vertex $u \in V(G) - X$ if and only if $G[\text{Epn}(v, X) \cup \{u, v\}]$ is complete.* ■

It is also possible to characterise minimal secure dominating sets. Let X be a dominating set of a graph G . Let $S'_G = \{v \in X \mid X - \{v\} \text{ is a dominating set of } G\}$ and define, for each $u \in V(G) - X$, the set $A_G(u, X) = \{v \in X \mid v \in N(u) \cap X \text{ and } (X - \{v\}) \cup \{u\} \text{ is a dominating set of } G\}$.

Theorem 3.7 (Cockayne *et al.* [32]) *A secure dominating set X of a graph G is minimal if and only if, for each $s \in S'_G$ with $N(s) \cap S'_G \neq \emptyset$, there exists $u_s \in V(G) - X$ such that, for each $v \in A_G(u_s, X) - \{s\}$, either*

1. *there exists $w \in V(G) - X$ such that $N(w) \cap X = \{v, s\}$ and $u_s \notin N(w)$, or*
2. *$N(s) \cap X = \{v\}$ and $u_s \in N(v) - N(s)$.* ■

3.2.1 Bounds on γ_s

Cockayne *et al.* [30] were able to relate the secure domination number $\gamma_s(G)$ of a graph G to its matching number $\nu(G)$, by establishing the following result.

Proposition 3.2 (Cockayne *et al.* [30]) *For any graph G of order n , $\gamma_s(G) \leq n - \nu(G)$.* ■

¹A clique covering of a graph G is set of cliques with the property that every vertex of G is a member of at least one clique. A minimum clique covering is a clique covering of minimum size, and the size of such a minimum clique covering of a graph G is known as the *clique covering number* of G , denoted by $\theta(G)$.

This upper bound was also found to be sharp for the infinite class of galaxies. Furthermore, Cockayne *et al.* [30] showed that $\gamma_s(G) \leq \frac{3}{2}\beta(G)$ if G is claw-free and that $\gamma_s(G) \leq \beta(G)$ if G is, in addition, also \mathcal{C}_5 -free. Upper bounds on $\gamma_s(G)$ were also obtained for a connected, claw-free graph G of order n in terms of the minimum degree δ of G . These bounds are $\gamma_s(G) \leq \frac{3n}{\delta+3}$ if G is claw-free, and $\gamma_s(G) \leq \frac{2n}{\delta+2}$ if G is also \mathcal{C}_5 -free.

Corollary 3.1 (Cockayne *et al.* [30]) *If a graph G of order n contains a perfect matching, then $\gamma_s(G) \leq n/2$. If a graph G of order n is connected and claw-free, then $\gamma_s(G) \leq \lceil n/2 \rceil$. ■*

The next two results provide lower bounds on the secure domination number of any graph G for \mathcal{K}_3 -free and \mathcal{K}_4 -free graphs in terms of the order and maximum degrees of G .

Theorem 3.8 (Cockayne *et al.* [32]) *For any triangle-free graph G of order n with maximum degree Δ ,*

$$\gamma_s(G) \geq \frac{n(2\Delta - 1)}{\Delta^2 + 2\Delta - 1}.$$

Furthermore, for each possible value of Δ , this bound is attained for infinitely many values of n . ■

Cockayne *et al.* [32] proved the above result by employing techniques from linear programming. A similar technique was used in the proof of the following result.

Theorem 3.9 (Cockayne *et al.* [32]) *For any \mathcal{K}_4 -free graph G of order n with maximum degree Δ ,*

$$\gamma_s(G) \geq \frac{n(2\Delta - 3)}{\Delta^2 + 2\Delta - 5}.$$

Furthermore, for each possible value of $\Delta \geq 3$, this bound is attained for infinitely many values of n . ■

The bounds in Theorems 3.8 and 3.9 were generalised as follows by Cockayne *et al.* [30].

Theorem 3.10 (Cockayne *et al.* [30]) *Let G be a graph of order n with maximum degree $\Delta \geq 3$. If G is \mathcal{K}_t -free for some $3 \leq t \leq \Delta + 1$, then*

$$\gamma_s(G) \geq \frac{n(2\Delta - 2t + 5)}{(\Delta + 1)^2 - (t - 1)(t - 2)}.$$

Furthermore, for all values of Δ and t satisfying the above hypothesis, the bound is attained for infinitely many values of n . ■

The following observation served as an important contribution to the work by Burger *et al.* [22].

Proposition 3.3 (Burger *et al.* [22]) *Let G_1 and G_2 be vertex-disjoint subgraphs of a graph G such that $V(G) = V(G_1) \cup V(G_2)$. If $S_1 \subseteq V(G_1)$ is a secure dominating set of G_1 and $S_2 \subseteq V(G_2)$ is a dominating set of G_2 , then every vertex in $V(G_1)$ is securely dominated by $S = S_1 \cup S_2$ in G . ■*

Burger *et al.* [22] were the first to note formally that adding edges to a graph does not increase the secure domination number of the graph.

Proposition 3.4 (Burger *et al.* [22]) *If G_1, G_2, \dots, G_r are vertex-disjoint subgraphs of a graph G such that every vertex of G belongs to exactly one of these subgraphs, then*

$$\gamma_s(G) \leq \sum_{i=1}^r \gamma_s(G_i).$$

■

Proposition 3.4 was instrumental in establishing the following important result.

Proposition 3.5 (Burger *et al.* [22]) *If G is a connected graph of order n with minimum degree at least two that is not a 5-cycle, then $\gamma_s(G) \leq n/2$ and this bound is sharp.*

■

The above result was established by covering a subset of the vertices of G by vertex disjoint copies of subgraphs, each of which is isomorphic to K_2 or to an odd cycle.

3.2.2 Special graph classes

Values for the parameter γ_s have been established for some graph classes, including the complete graph K_n for all $n \in \mathbb{N}$, the complete bipartite graph $K_{p,q}$ for all $p, q \in \mathbb{N}$, and the complete multi-partite graph K_{n_1, \dots, n_k} for all $n_i \in \mathbb{N}$ with $i = 1, \dots, k$ and $k \geq 3$.

Proposition 3.6 (Cockayne *et al.* [32]) *For any natural number n , $\gamma_s(K_n) = 1$.*

■

It is easy to see that complete graphs form the only infinite class of graphs which admit minimum secure dominating sets of cardinality 1.

Proposition 3.7 (Cockayne *et al.* [32]) *For any natural numbers p and q with $p \leq q$,*

$$\gamma_s(K_{p,q}) = \begin{cases} q & \text{if } p = 1 \\ 2 & \text{if } p = 2 \\ 3 & \text{if } p = 3 \\ 4 & \text{if } p \geq 4. \end{cases}$$

■

An important aspect in the proof of the above result is the realisation that in any complete bipartite graph $K_{p,q}$ with $2 \leq p \leq q$, a minimum secure dominating set can be formed by at most two vertices from each partite set.

Proposition 3.8 (Cockayne *et al.* [32]) *For any natural number $k \geq 3$ and natural numbers n_1, \dots, n_k satisfying $n_1 \leq n_2 \leq \dots \leq n_k$,*

$$\gamma_s(K_{n_1, \dots, n_k}) = \begin{cases} 2 & \text{if } n_1 = 1, n_2 \leq 2 \\ 2 & \text{if } n_1 = 2 \\ 3 & \text{otherwise.} \end{cases}$$

■

The proof of the above result hinges on the fact that it is possible to find a minimum secure dominating set of a complete k -partite graph which contains at most two vertices in a single partite set.

Cockayne *et al.* [32] determined the secure domination number of a path exactly.

Proposition 3.9 (Cockayne *et al.* [32]) For any natural number n ,

$$\gamma_s(\mathcal{P}_n) = \left\lceil \frac{3n}{7} \right\rceil. \quad \blacksquare$$

A similar result holds for cycles.

Proposition 3.10 (Cockayne *et al.* [32]) For any natural number $n \geq 4$,

$$\gamma_s(\mathcal{C}_n) = \left\lceil \frac{3n}{7} \right\rceil. \quad \blacksquare$$

An upper bound was also established by Cockayne *et al.* [32] on both the secure domination number and the weak Roman domination number of a grid graph in the plane.

Proposition 3.11 (Cockayne *et al.* [32]) For any natural numbers p and q ,

$$\gamma_r(\mathcal{P}_p \square \mathcal{P}_q) \leq \gamma_s(\mathcal{P}_p \square \mathcal{P}_q) \leq \left\lceil \frac{pq}{3} \right\rceil + 2. \quad \blacksquare$$

The result of Proposition 3.11 was used to prove the following result for grid graphs on a torus.

Corollary 3.2 (Cockayne *et al.* [32]) For any natural numbers $p, q \geq 3$,

$$\gamma_r(\mathcal{C}_p \square \mathcal{C}_q) \leq \gamma_s(\mathcal{C}_p \square \mathcal{C}_q) \leq \left\lceil \frac{pq}{3} \right\rceil. \quad \blacksquare$$

The following result hinges on the fact that the Cartesian product $\mathcal{C}_p \square \mathcal{C}_q$ has maximum degree 4 and is triangle-free; it is an immediate consequence of Theorem 3.8.

Corollary 3.3 (Cockayne *et al.* [32]) For any natural numbers p and q ,

$$\gamma_s(\mathcal{C}_p \square \mathcal{C}_q) \geq \left\lceil \frac{7pq}{23} \right\rceil. \quad \blacksquare$$

Secure domination of trees was studied by Cockayne [27]. The following bound on the secure domination number of a tree T was established in terms of the maximum degree of T .

Theorem 3.11 (Cockayne [27]) For any tree T of order n and maximum degree $\Delta \geq 3$,

$$\gamma_s(T) \geq \frac{\Delta n + \Delta - 1}{3\Delta - 1}.$$

For each possible value of $\Delta \geq 2$, this bound is attained for infinitely many values of n . \blacksquare

The above result was established using techniques similar to those in the proofs of Theorems 3.8–3.10.

The notion of excellence in graphs enabled Mynhardt *et al.* [79] to characterise trees with equal domination and secure domination numbers. A graph G is said to be γ -excellent if each vertex of G is contained in some minimum dominating set of G . Some useful properties of γ -excellent trees were established and used in the characterisation of (γ, γ_s) -trees — that is, trees T for which $\gamma(T) = \gamma_s(T)$.

3.2.3 Edge removal criticality in secure graph domination

In 2009, Grobler and Mynhardt [55] considered the effect of a single edge removal from a graph and characterised all graphs for which $\gamma_s(G - e) > \gamma_s(G)$, where $e \in E(G)$. This class of graphs is called γ_s -ER-critical graphs. They started by partitioning the vertices of a graph into various classes and then proceeded to characterise γ_s -ER critical graphs by means of a recursive construction process. They specialised their results by determining the classes of all bipartite γ_s -ER critical graphs and all γ_s -ER critical trees.

To describe their method of characterisation, let G be an arbitrary graph and let $X = \{x_1, \dots, x_k\}$ be a subset of the vertex set of G . Also define the sets

$$\begin{aligned} Z_i &= \text{Epn}(x_i, X) \text{ for each } x_i \in X, \\ P &= \bigcup_{i=1}^k Z_i, \text{ and} \\ Y &= V(G) - (X \cup P). \end{aligned}$$

The set X may be partitioned into the four subsets,

$$\begin{aligned} X_1 &= \{x \in X \mid N(x) \cap Y = \emptyset\}, \\ X_2 &= \{x \in X - X_1 \mid x \text{ does not defend any vertex in } Y\}, \\ X_3 &= \{x \in X - X_1 \mid x \text{ defends some but not all vertices in } N(x) \cap Y\}, \text{ and} \\ X_4 &= \{x \in X - X_1 \mid x \text{ defends all vertices in } N(x) \cap Y\}. \end{aligned}$$

The sets

$$\begin{aligned} U_i &= \{y \in Y \mid x_i \text{ uniquely defends } y\}, \\ U &= \bigcup_{i=1}^k U_i, \text{ and} \\ Y_{ij} &= \{y \in Y \mid x_i \text{ and } y_i \text{ jointly defends } y\}. \end{aligned}$$

are also required. Using the above sets, Grobler and Mynhardt [55] characterised the class of γ_s -ER-critical graphs as follows.

Theorem 3.12 (Grobler & Mynhardt [55]) *A graph G is γ_s -ER-critical if and only if, for every secure dominating set X of G ,*

1. X and Y are respectively independent;
2. every $y \in Y$ has exactly two neighbours in X ;
3. if $x_i \in X$ defends a vertex in Y (i.e. $x_i \in X_3 \cap X_4$), then $|U_i| \geq 2$;
4. the only edges in P are in $G[Z_i]$, where $x_i \in X$;
5. the only edges between Y and P are between Z_i and the vertices in Y that are defended by $x_i \in X$;
6. if $x_i \in X$ jointly defends a vertex in Y , then $Z_i = \emptyset$. ■

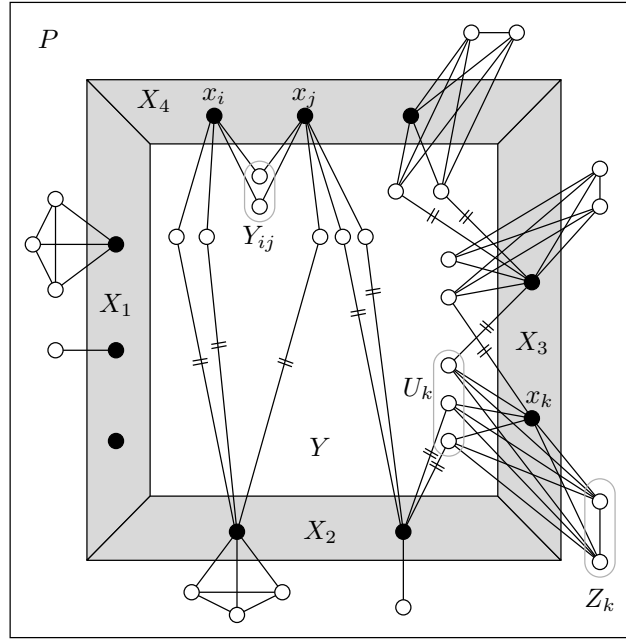

 FIGURE 3.3: A γ_s -set $X = X_1 \cup X_2 \cup X_3 \cup X_4$ in a γ_s -ER-critical graph.

Figure 3.3 contains a graphical representation of an γ_s -ER-critical graph. The symbol “=” on an edge x_iy ($y \in Y$) means that x_i does not defend y .

This result enabled Grobler and Mynhardt to construct the class \mathcal{G} of γ_s -ER-critical graphs recursively. The construction starts with a forest F consisting of $k \geq 1$ disjoint stars other than K_2 . The centres of these stars form the set $X = \{x_1, \dots, x_k\}$. At this stage of the construction, the set of leaves adjacent to x_i forms the set of vertices in U_i . The following four steps are used to add vertices and edges to F :

Step 1. For each $x_i \in X$, each vertex $u \in U_i$ is joined to exactly one vertex in $X - \{x_i\}$. Let

$$\begin{aligned} X_1 &= \{x_i \in X \mid U_i = \emptyset \text{ and } x_i \text{ is not adjacent to a vertex in } U\}, \\ X_2 &= \{x_i \in X \mid U_i = \emptyset \text{ and } x_i \text{ is adjacent to a vertex in } U\}, \\ X_3 &= \{x_i \in X \mid U_i \neq \emptyset \text{ and } x_i \text{ is adjacent to a vertex in } U - U_i\}, \text{ and} \\ X_4 &= \{x_i \in X \mid U_i \neq \emptyset \text{ and } x_i \text{ is not adjacent to a vertex in } U - U_i\}. \end{aligned}$$

Step 2. For each vertex $x_i \in X_2 \cup X_3$, a non-empty set Z_i of new vertices is added. Exactly those edges for which $G[\{x_i\} \cup Z_i]$ is complete and each vertex in Z_i is adjacent to each vertex in U_i are added.

Step 3. For each pair of distinct vertices $x_i, x_j \in X_4$, a (possibly empty) set Y_{ij} of new vertices is added, joining each vertex in Y_{ij} to x_i and x_j and no other vertices. Let

$$W = \{x_i \in X_4 \mid Y_{ij} \neq \emptyset \text{ for some } x_j \in X_4 - \{x_i\}\}.$$

The vertices in W therefore form a subset of the vertices in X_4 , such that each vertex in W does not jointly defend any vertex.

Step 4. For each $x_i \in X_1 \cup (X_4 - W)$, a (possibly empty) set Z_i of new vertices is added together with only those edges necessary to ensure that $G[\{x_i\} \cup Z_i]$ is complete and that each vertex in Z_i is adjacent to each vertex in U_i . Note that for each vertex $x_i \in X_1$, the set U_i is empty, and the only edges that have to be added are the edges ensuring that $G[\{x_i\} \cup Z_i]$ is a clique.

Using the result of Theorem 3.12, Grobler and Mynhardt [55] established the following characterisation by a series of technical arguments.

Theorem 3.13 (Grobler & Mynhardt [55]) *A graph G is γ_s -ER-critical if and only if G is a member of the class \mathcal{G} whose construction is described above. ■*

Figure 3.4 contains graphical illustrations of the fourteen non-isomorphic γ_s -ER-critical of order 6. These graphs were constructed using the result of Theorem 3.13. Note that the first ten graphs in the figure (Figure 3.4(a)–(j)) are all unions of cliques.

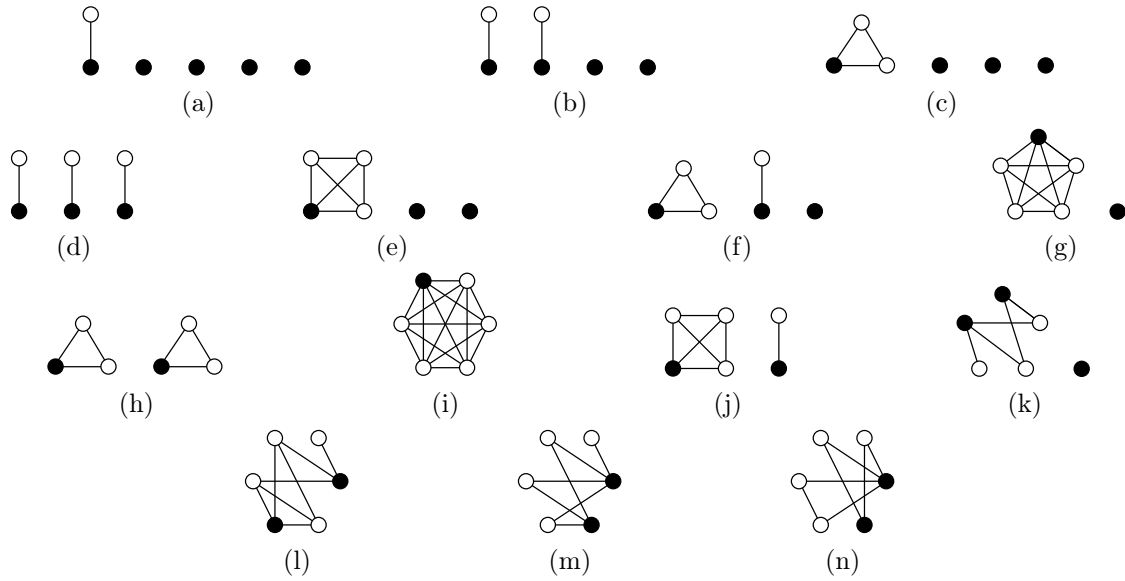


FIGURE 3.4: All γ_s -ER-critical graphs of order 6.

Figure 3.5 contains, as an example, a graphical illustration of the steps of constructing the γ_s -ER-critical graph in Figure 3.4(f). The construction starts with three stars of the form $\mathcal{K}_{1,0}$ (three isolated vertices), namely x_1 , x_2 and x_3 and thus $U_i = \emptyset$ for all $i = 1, 2, 3$. It follows that $X_1 = \{x_1, x_2, x_3\}$. No edges are added in Steps 1–3 of the construction process. In Step 4, x_1 receives two external private neighbours in the set Z_1 and edges are added such that $G[\{x_1\} \cup Z_1]$ is complete. Similarly, x_2 receives an external private neighbour in the set Z_2 and an edge is added to ensure that $G[\{x_2\} \cup Z_2]$ is a clique, as shown in Figure 3.5(b).



FIGURE 3.5: Constructing the γ_s -ER-critical graph $\mathcal{K}_3 \cup \mathcal{K}_2 \cup \mathcal{K}_1$ in Figure 3.4(f). The construction starts with a forest of three disjoint stars $3\mathcal{K}_{1,0}$, for which the centres x_1 , x_2 and x_3 are shown in Figure 3.6(a). Since $U_i = \emptyset$ for all $i = 1, 2, 3$, it follows that $X_1 = \{x_1, x_2, x_3\}$. (b) In Step 4 of the construction process, x_1 receives two external private neighbours in the set Z_1 and edges are added such that $G[\{x_1\} \cup Z_1]$ is complete. Similarly, x_2 receives an external private neighbour in the set Z_2 and an edge is added to ensure that $G[\{x_2\} \cup Z_2]$ is complete.

Figure 3.6 contains another example illustrating the construction process described above — this time for the γ_s -ER-critical graph in Figure 3.4(l). The construction starts with two disjoint stars $\mathcal{K}_{1,0} \cup \mathcal{K}_{1,2}$ with centres x_1 and x_2 , as shown in Figure 3.6(a). It follows that $X_4 = \{x_1\}$ and $X_2 = \{x_2\}$. In Step 1 of the construction process the vertex x_2 is therefore joined to both the vertices in U_1 , as shown in Figure 3.6(b). In Step 2, x_2 receives an external private neighbour in the set Z_2 and an edge is added to ensure that $G[\{x_2\} \cup Z_2]$ is complete, as shown in Figure 3.6(c). In Step 4, x_1 receives an external private neighbour in the set Z_1 and edges are added so that $G[\{x_1\} \cup Z_1]$ is complete and so that the vertex in Z_1 is adjacent to each vertex in U_1 , as shown in Figure 3.6(d).

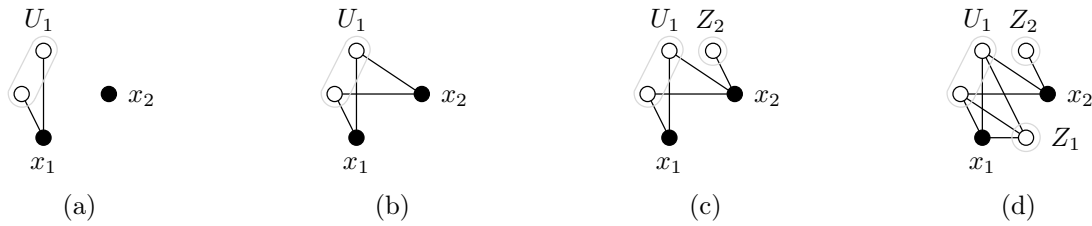


FIGURE 3.6: Constructing the γ_s -ER-critical graph in Figure 3.4(l). (a) The construction starts with two disjoint stars $\mathcal{K}_{1,0} \cup \mathcal{K}_{1,2}$ with centres x_1 and x_2 . Since $|U_1| > 0$ and $|U_2| = 0$, it follows that $X_4 = \{x_1\}$ and $X_2 = \{x_2\}$. (b) In Step 1, x_2 is joined to both the vertices in U_1 . (c) In Step 2, x_2 receives an external private neighbour in the set Z_2 and an edge is added to ensure that $G[\{x_2\} \cup Z_2]$ is complete. (d) In Step 4, x_1 receives an external private neighbour in the set Z_1 and edges are added so that $G[\{x_1\} \cup Z_1]$ is complete and so that the vertex in Z_1 is adjacent to each vertex in U_1 .

A number of topics for future research were also proposed by Grobler and Mynhardt [55]. One of these topics for future research is to determine the smallest number of arbitrary edges to be removed from a graph to ensure that the secure domination number necessarily increases.

3.3 Generalised graph protection parameters

The notion of secure domination may be generalised in various ways. A natural generalisation is that a dominating set may not merely be sought after one move, but rather after each of $k \geq 1$ moves. Another natural generalisation is to allow more than one guard to move simultaneously in order to afford some guard the opportunity to deal with an attack at an unoccupied vertex. Some general properties of these generalised domination parameters are outlined in this section.

3.3.1 Finite higher order generalisations

The notion of weak Roman and secure domination was generalised by Burger *et al.* [12] so that safe guard configurations are guaranteed after each of $k \geq 1$ moves. To cater for the protection of a graph against a sequence of consecutive attacks, a superscript is introduced in the notation of a guard function, indicating the number of attacks already defended against. For some integer $i \in \mathbb{N}$, let $f^{(i)} = (V_0^{(i)}, V_1^{(i)}, \dots)$ be a guard function of G and let $v_i \in V(G)$. Let $f^{(i+1)}$ denote another guard function formed from $f^{(i)}$ and v_i . If $v_i \in V_0^{(i)}$, then $f^{(i+1)}$ is the guard function obtained from $f^{(i)}$ by the movement of a guard from its position at $u_i \in N[v_i] \cap (V(G) - V_0^{(i)})$ along an edge to v_i in response to an attack. However, if $v_i \in V(G) - V_0^{(i)}$, then no movement

is necessary since u_i is the vertex v_i , and $f^{(i+1)} = f^{(i)}$. Formally, for $s \in V(G)$,

$$\begin{aligned} f^{(i+1)}(s) &= \text{move}(f^{(i)}(s), u_i \rightarrow v_i) \\ &:= \begin{cases} f^{(i)}(s) - 1 & \text{if } s = u_i \text{ and } v_i \in V_0^{(i)} \\ 1 & \text{if } s = v_i \text{ and } v_i \in V_0^{(i)} \\ f^{(i)}(s) & \text{if } s \in V(G) - \{u_i, v_i\} \text{ or } v_i \notin V_0^{(i)}. \end{cases} \end{aligned}$$

Successful defense against a sequence of attacks at vertices v_0, v_1, \dots, v_{k-1} , starting with the guard function $f^{(0)}$, is a recursive process. For each sequential attack $i = 0, 1, 2, \dots, k-1$, if $v_i \in V_0^{(i)}$, a guard is to be moved along an edge from $u_i \in V(G) - V_0^{(i)}$ so that $f^{(i+1)}(s) = \text{move}(f^{(i)}(s), u_i \rightarrow v_i)$ is a safe guard function of G . The existence of a sequence u_0, u_1, \dots, u_{k-1} satisfying the above conditions is required for successful defense [4].

A k -secure dominating function is a safe guard function $f^{(0)} = (V_0^{(0)}, V_1^{(0)})$ with the property that, for any sequence of vertices v_0, v_1, \dots, v_{k-1} , there exists a sequence of vertices $u_i \in V_1^{(i)}$ such that the functions $f^{(i+1)}(s) = \text{move}(f^{(i)}(s), u_i \rightarrow v_i)$ are also safe guard functions for all $i = 0, \dots, k-1$. The minimum weight of a k -secure dominating function is denoted by $\gamma_{s,k}(G)$, which is called the k -secure dominating number of G .

The following growth relationships were first noted by Burger *et al.* [12].

Theorem 3.14 (Burger *et al.* [12]) For any graph G and any $k \in \mathbb{N}_0$, $\gamma_{s,k}(G) \leq \gamma_{s,k+1}(G)$. ■

The next result, also by Burger *et al.* [12], describes how edge removal from a graph effects these domination numbers.

Theorem 3.15 (Burger *et al.* [12]) For any graph G and any edge $e \in E(G)$, $\gamma_{s,k}(G) \leq \gamma_{s,k}(G - e)$ for all $k \in \mathbb{N}_0$. ■

Burger *et al.* [12] also determined the values of $\gamma_{s,k}(G)$ for special graph classes G . A further generalisation by Benecke [4] allows for an arbitrary number of guards to be stationed at a vertex.

Let $k, \ell \in \mathbb{N}$. A k^{th} -order ℓ -dominating function $((\ell, k)$ -DF) of a graph G is a safe guard function $f^{(0)} = (V_0^{(0)}, V_1^{(0)}, \dots, V_\ell^{(0)})$ with the property that, for any sequence of vertices v_0, v_1, \dots, v_{k-1} , there exists a sequence of vertices $u_i \in N[v_i] \cap (V(G) - V_0^{(i)})$, $i = 0, 1, \dots, k-1$ such that the guard functions $f^{(i+1)} = \text{move}(f^{(i)}, u_i \rightarrow v_i)$ are safe guard functions for all $i = 0, 1, \dots, k-1$. The minimum weight of an (ℓ, k) -DF is denoted by

$$\gamma_{\ell,k}(G) = \min_{(\ell,k)\text{-DF}} \left(\sum_{j=1}^{\ell} j |V_j^{(0)}| \right)$$

and is called the k^{th} -order ℓ -dominating number of G .

If $f^{(0)} = (V_0^{(0)}, V_1^{(0)}, \dots, V_\ell^{(0)})$ is an (ℓ, k) -DF, then $|V_j^{(i)}| \geq |V_j^{(i+1)}|$ for all $i = 0, 1, \dots, k-1$ and $j = 2, 3, \dots, \ell$, implying that the number of guards on an already occupied vertex can never increase as a result of a guard movement. In addition to the definition, the case $k = 0$ is allowed as a special convention. In this case there are no problem vertices and hence the configuration $f^{(0)}$ remains static (*i.e.* there are no moves), which means that $f^{(0)}$ must be a dominating function in the classical sense. Hence $\gamma(G) = \gamma_{1,0}(G)$ for any graph G , while $\gamma_s(G) = \gamma_{1,1}(G)$.

Proposition 3.12 (Benecke [4]) For any graph G and any $k, \ell \in \mathbb{N}$,

$$\gamma_{\ell+1,k}(G) \leq \gamma_{\ell,k}(G) \quad \text{and} \quad \gamma_{\ell,k}(G) \leq \gamma_{\ell,k+1}(G).$$

■

3.3.2 Generalisations to include infinitely many moves

A further variation on the theme of secure domination was suggested by Burger *et al.* [13], for the case when perpetual or eternal security in a graph is required. A ∞ -secure dominating function is a k -secure dominating function in the limit as $k \rightarrow \infty$. The minimum weight of an ∞ -secure dominating function is denoted by $\gamma_{s,\infty}(G) = \lim_{k \rightarrow \infty} \gamma_{s,k}(G)$, and is called the ∞ -secure dominating number of G . Burger *et al.* [12, 13] established the following inequality chain in 2004.

Theorem 3.16 (Burger *et al.* [13]) For any graph G , $\gamma(G) \leq \gamma_{s,k}(G) \leq \gamma_{s,\infty}(G) \leq \chi(\overline{G})$.

■

Burger *et al.* [13] went on to provide values for $\gamma_{s,\infty}(G)$ for special graph classes G , including paths, cycles, multipartite graphs, hexagonal graphs and the cartesian products of complete graphs, cycles and paths. Goddard *et al.* [51] were able to show that for a graph G with independence number equal to two, $\gamma_{s,\infty}(G) \leq 3$. They conjectured that there is a constant c such that $\gamma_{s,\infty}(G) \leq c$ for all graphs with $\beta(G) = 3$. In 2007, Klostermeyer and MacGillivray [66] established the following result.

Theorem 3.17 (Klostermeyer & MacGillivray [66]) For any graph G with independence number β , $\gamma_{s,\infty}(G) \leq \binom{\beta+1}{2}$.

■

Golwasser and Klostermeyer [52] showed in 2008 that there are graphs G for which $\gamma_{s,\infty}(G) \geq \binom{\beta+1}{2}$, showing that the bound in Proposition 3.17 is tight. Klostermeyer and Mynhardt [70] studied a variant of this problem in which the configuration of guards induce a total dominating set.

3.3.3 Generalisations involving multiple moves

In 2005, Goddard *et al.* [51] focused on security problems where multiple guards can move simultaneously in response to an attack, which they called m -security. For some integer $i \in \mathbb{N}$, let $f^{(i)} = (V_0^{(i)}, V_1^{(i)}, \dots)$ be a guard function of G . Let $f^{(i+1)}$ denote another guard function formed from $f^{(i)}$, where each vertex $v \in V(G)$ can move to a vertex in $N(v)$, such that $f^{(i+1)}$ is a safe guard function of G .

A *eternal m -security function* is a safe guard function $f^{(0)} = (V_0^{(0)}, V_1^{(0)})$ with the property that for any sequence of vertices v_0, v_1, \dots, v_{k-1} , such that there exists guard functions $f^{(i+1)}$, where each vertex $v_i \in V(G)$ can move along an edge to a vertex in $N(v_i)$, which are safe guard functions for all $i = 0, 1, \dots, k-1$. The minimum weight of an eternal m -security function is denoted by $\gamma_{s,m}^*(G)$ and is called the *eternal m -security number* of G .

The eternal m -security number of a graph is related as follows to its domination number and its independence number.

Theorem 3.18 (Goddard *et al.* [51]) For any graph G , $\gamma(G) \leq \gamma_{s,m}^*(G) \leq \beta(G)$.

■

Goddard *et al.* [51] were able to establish bounds on the eternal m -security number of an arbitrary graph and also determined the eternal m -security number for the infinite classes of complete graphs, complete bipartite graphs, paths and cycles.

A *colonisation* of a graph G is a partition of the vertex set of G into subgraphs each containing a *dominator*, that is, a vertex adjacent to all other vertices in the subgraph. The weight of a colonisation of G counts 1 for each clique and 2 for each non-clique, and the minimum weight of a colonisation of G is denoted by $\theta_c(G)$.

Theorem 3.19 (Goddard *et al.* [51]) *For any graph G , $\gamma_{s,m}^*(G) \leq \theta_c(G)$.* ■

Klostermeyer and MacGillivray [67] used the result in Theorem 3.19 to show that for any tree T , it holds that $\gamma_{s,m}^*(T) = \theta_c(T)$. In 2011, Klostermeyer and Mynhardt [69] focussed on graphs with equal eternal vertex cover and eternal domination numbers. The notion of an *eternal vertex cover* is similar to that of an eternal m -security [69], differing only in the sense that attacks take place on the edges of a graph. During an attack on an edge uv of a graph, at least one guard moves from u to v or from v to u to deal with the attack. The *eternal vertex cover number* of a graph G is the minimum cardinality of an eternal vertex cover and is denoted by $\alpha_m(G)$.

Theorem 3.20 (Klostermeyer & Mynhardt [69]) *For any connected graph $G \not\cong C_4$ with minimum degree greater than two, $\gamma_{s,m}^*(G) < \alpha_m(G)$.* ■

A *vertex cover* of a graph G is a set $S \subseteq V(G)$ such that for each edge $uv \in E(G)$, at least one of u or v is in S . The size of a minimum vertex cover of a graph G is denoted by $\alpha(G)$. In 2012, Klostermeyer and Mynhardt [71] established the following result.

Theorem 3.21 (Klostermeyer & Mynhardt [71]) *For any connected graph G with minimum degree greater than two and girth at least nine, $\gamma_{s,m}^*(G) < \alpha(G)$.* ■

3.4 Chapter summary

A survey of the literature on topics related to the protection of graphs was conducted in this chapter. Numerous graph protection parameters were reviewed in §3.1. More specifically, definitions of (classical) domination (formally introduced in §2.2.2), total domination, Roman domination, weak Roman domination and secure domination were given and some of the fundamental results involving these graph parameters were discussed. The five parameters mentioned above represent the minimum number of guards required to protect a graph against attacks under different conditions (*i.e.* for different definitions of the notion of “protection”). The first three parameters are applicable in a static protection framework, while the latter two apply to dynamic protection strategies involving guard moves. An important inequality chain involving these parameters may be found in Theorem 3.4.

A variety of results related to secure graph domination was presented in §3.2. A number of general bounds on the secure domination number were reviewed; these bounds will be used in the remainder of this dissertation. Furthermore, results on the exact values on the secure domination number were also reviewed for certain infinite classes of graphs. A number of bounds were also described for infinite classes of graphs for which exact values were not attainable. The notion of edge removal in secure graph domination was considered in §3.2.3, specifically with a focus on criticality in secure graph domination.

A number of variations on the notion of secure graph domination were considered in §3.3. Finite higher-order generalisations were considered in §3.3.1 and apply to the protection of a graph where more than a single attack occurs in the graph. The k -secure dominating function deals with protecting the graph against a sequence of attacks on k distinct vertices. Generalisations were also considered for the cases where a vertex may contain more than one guard. The case where an infinite sequence of attacks occurs was considered in §3.3.2, and this was followed by generalisations allowing the simultaneous movement of multiple guards during an attack in §3.3.3.

CHAPTER 4

Properties of minimum secure dominating sets

Contents

4.1 The structure of a secure dominating set	65
4.2 Graphs with small secure domination numbers	66
4.3 On the defenders in a minimum secure dominating set	72
4.4 Secure domination is NP -complete	75
4.5 Chapter summary	77

A number of novel results on the nature and computation of minimum secure dominating sets are presented in this chapter. The chapter opens with a description of three requirements for testing algorithmically whether a given subset of vertices of a graph G is a secure dominating set of G in §4.1. This result facilitates the characterisation of the classes of graphs G for which $\gamma_s(G) = 1, 2$ or 3 in §4.2. It is shown in §4.3 that every graph of minimum degree at least two possesses a minimum secure dominating set in which all vertices are defenders. The chapter closes in §4.4 with a proof that the decision problem associated with secure domination is **NP-complete**.

4.1 The structure of a secure dominating set

Let X be any subset of the vertex set $V(G)$ of a graph G . Then $V(G)$ can be partitioned into five subsets with respect to X . Denote by X_P the set of vertices in X which have private neighbours external to X , and let $P_X = \bigcup_{v \in X_P} \text{Epn}(v, X)$ be the set of these external private neighbours of the vertices in X_P . Furthermore, define $X_D = X - X_P$ and let D_X be the set of vertices in $V(G) - (X \cup P_X)$ that are dominated by X_D . Finally, let U_X be the set of vertices not in X , P_X or D_X . Then

$$V(G) = \underbrace{X_P \cup X_D}_X \cup P_X \cup U_X \cup D_X, \quad (4.1)$$

as illustrated in Figure 4.1. It follows by Theorem 3.6 that each vertex in D_X is defended by at least one vertex in X_D . Whilst every vertex in P_X is further *dominated* by exactly one vertex in X_P , some of the vertices in P_X may not be *defended* by any vertices in X_P . Even worse, some vertices in U_X may not even be dominated by any vertices in X , let alone be defended. For example, in Figure 4.1 the vertices $u_1, u_2 \in P_X$ are not defended (by v_1), because $G[\{v_1, u_1, u_2\}]$ is not complete. The vertices $u_3, u_4 \in P_X$ are, however, defended (by v_2), because $G[\{v_2, u_3, u_4\}]$ is complete. The vertices $u_5, u_6, u_7 \in P_X$ are similarly defended (by v_3, v_4 and v_4 , respectively). Furthermore, the vertex $u_8 \in U_X$ is defended (by v_4) since $G[\{v_4, u_6, u_7, u_8\}]$

is complete. However, $u_9 \in U_X$ is defended by neither v_3 nor v_4 , because both $G[\{v_3, u_5, u_9\}]$ and $G[\{v_4, u_6, u_7, u_9\}]$ are incomplete. Finally, $u_{10}, u_{11}, u_{12} \in U_X$ are not even dominated by X , let alone defended.

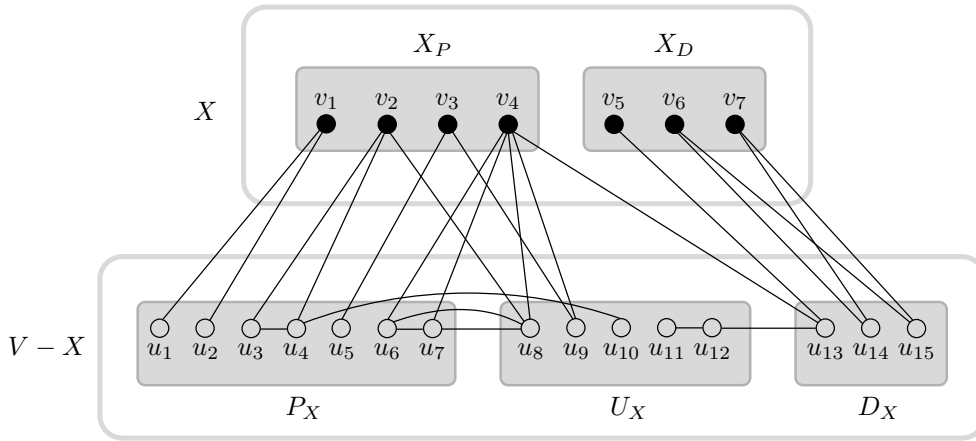


FIGURE 4.1: An example of the vertex set partition in (4.1).

In order to decide whether X is a secure dominating set of G , it suffices, by Theorem 3.6, to verify that:

- I every vertex in U_X is dominated by at least one vertex in X ,
- II the private neighbours in P_X of each vertex in X_P form a clique in G , and
- III there exists, for each $u \in U_X$, a vertex $v \in X$ such that $G[\text{Epn}(v, X) \cup \{u, v\}]$ is complete.

Note that there are no edges between vertices in X_D and vertices in U_X . Furthermore, if at least one of the above requirements I–III does not hold, then X is not a secure dominating set of G . These criteria have been ordered in increasing order of computational complexity above so as to represent an efficient set of secure domination testing criteria.

4.2 Graphs with small secure domination numbers

In this section, classes of graphs that have secure domination numbers 1, 2 or 3 are characterised. The simplest case is considered first — graphs with secure domination number 1.

Theorem 4.1 (Characterisation of graphs with secure domination number 1)

A graph has secure domination number 1 if and only if it is a complete graph.

Proof: Clearly, $\gamma_s(K_n) = 1$ by Proposition 3.6. Conversely, suppose G is a graph of order $n \geq 3$ that is not complete, but suppose, to the contrary, that $\gamma_s(G) = 1$ and let $X = \{x\}$ be a dominating set of G for some $x \in V(G)$. Since G contains two non-adjacent vertices u and v , it follows that $x \notin \{u, v\}$, for otherwise X would not be dominating. But then neither u nor v is defended by x , a contradiction. Therefore, $\gamma_s(G) \geq 2$. ■

In order to characterise graphs with secure domination number 2, the following graph construction is required. Let i, j be positive integers and let k, ℓ be non-negative integers. Furthermore,

let $\Phi(i, j, k, \ell)$ denote the graph of order $i + j + k + \ell$ and size $\binom{i}{2} + \binom{j}{2} + k(i + 1) + \ell(j + 1)$ containing two vertex-disjoint cliques \mathcal{K}_i and \mathcal{K}_j of orders i and j , respectively, together with two further disjoint vertex subsets \mathcal{U}_k and \mathcal{Y}_ℓ of vertices of cardinalities k and ℓ , respectively, to which the following edges are added:

1. Each vertex in \mathcal{K}_i is joined to all vertices of \mathcal{U}_k (if $\mathcal{U}_k \neq \emptyset$).
2. Each vertex in \mathcal{K}_j is joined to all vertices of \mathcal{Y}_ℓ (if $\mathcal{Y}_\ell \neq \emptyset$).
3. Some vertex $x \in V(\mathcal{K}_i)$ is joined to all vertices in \mathcal{Y}_ℓ (if $\mathcal{Y}_\ell \neq \emptyset$), and some vertex $y \in V(\mathcal{K}_j)$ is joined to all vertices in \mathcal{U}_k (if $\mathcal{U}_k \neq \emptyset$).

Note that no two vertices of \mathcal{U}_k are adjacent, and similarly for \mathcal{Y}_ℓ . The construction of $\Phi(i, j, k, \ell)$ is illustrated in Figure 4.2.

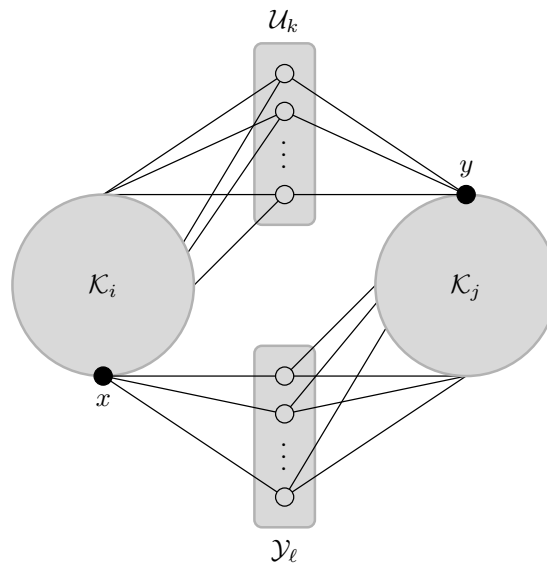


FIGURE 4.2: The graph $\Phi(i, j, k, \ell)$.

Theorem 4.2 (Characterisation of graphs with secure domination number 2)

A graph has secure domination number 2 if and only if it is not complete and contains $\Phi(i, j, k, \ell)$ in Figure 4.2 as spanning subgraph for some integers $i, j \geq 1$ and $k, \ell \geq 0$.

Proof: The set $\{x, y\}$ is clearly a secure dominating set of cardinality 2 for $\Phi(i, j, k, \ell)$. Since $\gamma_s(\Phi(i, j, k, \ell)) \neq 1$ by Theorem 4.1, it follows that $\gamma_s(\Phi(i, j, k, \ell)) = 2$, which settles the sufficiency.

For the necessity, suppose $X = \{x, y\}$ is a secure dominating set of some graph G of order n and consider the partition in (4.1). There are exactly three cases to consider in terms of the possible structure of G .

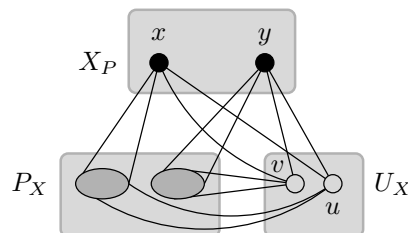


FIGURE 4.3: A spanning subgraph of G in Case 1 of the proof of Theorem 4.2.

Case 1: $x, y \in X_P$. In this case both X_D and D_X in Figure 4.1 are empty and the vertices in U_X are defended by vertices in X_P . If x defends some vertex $u \in U_X$, then $G[\text{Epn}(x, X_P) \cup \{x, u\}]$ forms a clique in G by Theorem 3.6. Similarly, if y defends some vertex $v \in U_X$, then $G[\text{Epn}(y, X_P) \cup \{y, v\}]$ forms a clique in G , as depicted in Figure 4.3. Let \mathcal{U}_k be the set of $k \geq 0$ vertices that form a clique, \mathcal{K}_i (say), together with x and its private neighbours, and let \mathcal{Y}_ℓ be the set of $\ell \geq 0$ vertices that form a clique, \mathcal{K}_j (say), together with y and its private neighbours. Note that \mathcal{U}_k and/or \mathcal{Y}_ℓ may possibly be empty. In this case G therefore contains the graph $\Phi(i, j, k, \ell)$ in Figure 4.2 as spanning subgraph (where $i + j + k + \ell = n$).

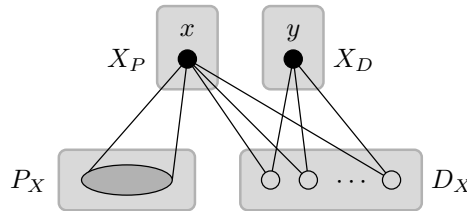


FIGURE 4.4: A spanning subgraph of G in Case 2 of the proof of Theorem 4.2.

Case 2: $x \in X_P$ and $y \in X_D$. In this case $U_X = \emptyset$ in Figure 4.1 and each vertex in D_X is adjacent to both $x \in X_P$ and $y \in X_D$, as depicted in Figure 4.4. Furthermore, the private neighbours of x form a clique, \mathcal{K}_i (say), in G together with x by Theorem 3.6. In this case, therefore, $\mathcal{U}_k = \emptyset$, \mathcal{K}_j contains only the vertex y , and \mathcal{Y}_ℓ contains all the vertices in D_X in Figure 4.2. Hence G contains the graph $\Phi(i, 1, 0, \ell)$ in Figure 4.2 as spanning subgraph (where $i + \ell + 1 = n$).

Case 3: $x, y \in X_D$. In this case both P_X and U_X in Figure 4.1 are empty, as depicted in Figure 4.5. Therefore, \mathcal{K}_i contains only the vertex x , \mathcal{K}_j contains only the vertex y , and $D_X = \mathcal{U}_k \cup \mathcal{Y}_\ell$ in Figure 4.1 (where $k + \ell + 2 = n$). Therefore, G contains the graph $\Phi(1, 1, k, \ell)$ in Figure 4.2 as spanning subgraph. ■

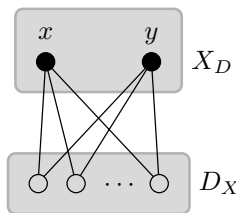
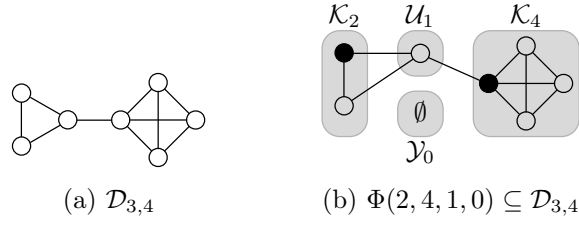


FIGURE 4.5: A spanning subgraph of G in Case 3 of the proof of Theorem 4.2.

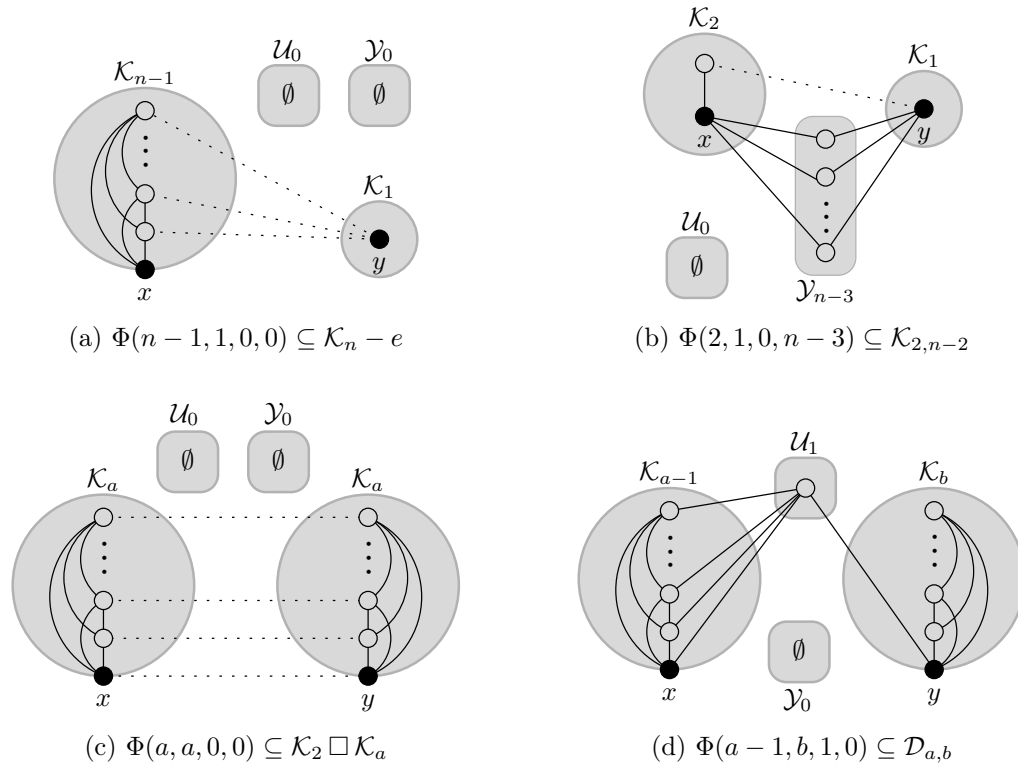
The characterisation in Theorem 4.2 may be used to prove succinctly that $\gamma_s(G) = 2$ for an incomplete graph that contains $\Phi(i, j, k, \ell)$ as spanning subgraph, by merely citing the parameters of the spanning subgraph $\Phi(i, j, k, \ell)$ as certificate. Note that multiple certificates may exist showing that $\gamma_s(G) = 2$ for a graph or graph class G .

Consider, as an example, the connected *dumbbell graph* $\mathcal{D}_{a,b}$ of order $a + b$ and size $\binom{a}{2} + \binom{b}{2} + 1$ obtained by joining two vertex disjoint cliques of orders a and b by a single edge. A graphical presentation of the dumbbell graph $\mathcal{D}_{3,4}$ is shown in Figure 4.6(a), while it is illustrated in Figure 4.6(b) that $\Phi(2, 4, 1, 0)$ is a spanning subgraph of $\mathcal{D}_{3,4}$.

Since $\Phi(a - 1, b, 1, 0)$ is a spanning subgraph of the dumbbell graph $\mathcal{D}_{a,b}$ in general, it follows immediately from Theorems 4.1 and 4.2 that $\gamma_s(\mathcal{D}_{a,b}) = 2$ for all $a, b \in \mathbb{N}_0$. Further examples of certificates showing that $\gamma_s(G) = 2$ for various infinite classes of graphs G are shown in Table 4.1. Figure 4.7 contains graphical representations of four of the certificates listed in Table 4.1 in the context of the full graph classes in the table.


 FIGURE 4.6: The dumbbell graph $\mathcal{D}_{3,4}$.

Graph class	Certificate
The complete graph less an edge, $\mathcal{K}_n - e$	$\Phi(n - 1, 1, 0, 0)$
The complete graph less two edges, $\mathcal{K}_n - 2e$	$\Phi(n - 2, 2, 0, 0)$
The complete bipartite graph, $\mathcal{K}_{2,n-2}$	$\Phi(2, 1, 0, n - 3)$
The book graph $\mathcal{B}_n = \mathcal{K}_2 + \mathcal{K}_{n-2}$	$\Phi(2, 1, 0, n - 3)$
The cartesian product, $\mathcal{K}_2 \square \mathcal{K}_a$	$\Phi(a, a, 0, 0)$
The dumbbell graph, $\mathcal{D}_{a,b}$	$\Phi(a - 1, b, 1, 0)$

 TABLE 4.1: Certificates showing that $\gamma_s(G) = 2$ for various infinite classes of graphs G .

 FIGURE 4.7: Various infinite graph classes together with certificates showing that these graphs have secure domination number 2. Dotted lines represent edges that are present in the graph classes of Table 4.1, but not in the certificates listed in the table. (a) The complete graph less an edge. (b) The complete bipartite graph. (c) The cartesian product $\mathcal{K}_2 \square \mathcal{K}_a$. (d) The dumbbell graph.

This section is concluded by characterising graphs G for which $\gamma_s(G) = 3$. The following graph construction is required for this characterisation. Let i, j, k be positive integers and let r, s, t be non-negative integers. Let $\Psi(i, j, k, r, s, t)$ denote the graph of order $i + j + k + r + s + t$ and size $\binom{i}{2} + \binom{j}{2} + \binom{k}{2} + r(i + 1) + s(j + 1) + t(k + 1)$ containing three vertex-disjoint cliques \mathcal{K}_i , \mathcal{K}_j and

\mathcal{K}_k of orders i, j and k , respectively, together with three further disjoint vertex subsets $\mathcal{U}_r, \mathcal{Y}_s$ and \mathcal{Z}_t of cardinalities r, s and t , respectively, to which the following edges are added:

1. Each vertex in \mathcal{U}_r is joined to all vertices in \mathcal{K}_i (if $\mathcal{U}_r \neq \emptyset$).
2. Each vertex in \mathcal{Y}_s is joined to all vertices in \mathcal{K}_j (if $\mathcal{Y}_s \neq \emptyset$).
3. Each vertex in \mathcal{Z}_t is joined to all vertices in \mathcal{K}_k (if $\mathcal{Z}_t \neq \emptyset$).
4. Vertices $x \in V(\mathcal{K}_i), y \in V(\mathcal{K}_j)$ and $z \in V(\mathcal{K}_k)$ are chosen, joining
 - (a) each vertex in \mathcal{U}_r to either y or z (but not both), if $\mathcal{U}_r \neq \emptyset$.
 - (b) each vertex in \mathcal{Y}_s to either x or z (but not both), if $\mathcal{Y}_s \neq \emptyset$.
 - (c) each vertex in \mathcal{Z}_t to either x or y (but not both), if $\mathcal{Z}_t \neq \emptyset$.

Note that no two vertices in \mathcal{U}_r are adjacent, and similarly for \mathcal{Y}_s and \mathcal{Z}_t . The construction of the graph $\Psi(i, j, k, r, s, t)$ is illustrated in Figure 4.8.

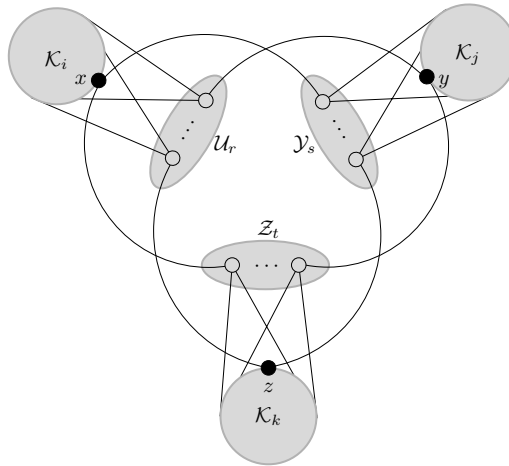


FIGURE 4.8: The graph $\Psi(i, j, k, r, s, t)$.

Theorem 4.3 (Characterisation of graphs with secure domination number 3)

An incomplete graph has secure domination number 3 if and only if it does not contain the graph $\Phi(i, j, k, \ell)$ in Figure 4.2 as subgraph for any integers $i, j \geq 1$ and $k, \ell \geq 0$, but contains the graph $\Psi(i, j, k, r, s, t)$ in Figure 4.8 as spanning subgraph for some integers $i, j, k \geq 1$ and $r, s, t \geq 0$.

Proof: The set $\{x, y, z\}$ is clearly a secure dominating set of cardinality 3 for the graph $\Psi(i, j, k, r, s, t)$. Furthermore, $\gamma_s(\Psi(i, j, k, r, s, t)) \neq 1, 2$ by Theorems 4.1 and 4.2. It follows that $\gamma_s(\Psi(i, j, k, r, s, t)) = 3$, which settles the sufficiency.

For the necessity, suppose $X = \{x, y, z\}$ is a secure dominating set of some graph G of order n and again consider the partition in (4.1). There are exactly four cases to consider in terms of the possible structure of G .

Case 1: $x, y, z \in X_P$. In this case both X_D and D_X in Figure 4.1 are empty and the vertices in U_X are defended by vertices in X_P . Therefore, each vertex in U_X is adjacent to at least two vertices in X_P , as depicted in Figure 4.9. Let \mathcal{U}_r be the set of $r \geq 0$ vertices that form a clique, \mathcal{K}_i (say), together with x and its private neighbours according to Theorem 3.6. Similarly, let \mathcal{Y}_s be the set of $s \geq 0$ vertices that form a clique, \mathcal{K}_j (say), together with y and its private neighbours, and let \mathcal{Z}_t be the set of $t \geq 0$ vertices that form a clique, \mathcal{K}_k (say), together with z and its private neighbours. Note that $\mathcal{U}_r, \mathcal{Y}_s$ and/or \mathcal{Z}_t may possibly be empty. Since the vertices in \mathcal{U}_r are defended by x and each vertex in U_X is defended by at least two vertices in X_P , it follows that all the vertices in \mathcal{U}_r are adjacent to y or z . Similarly, all the vertices in \mathcal{Y}_s are adjacent to

x or z , and all the vertices in \mathcal{Z}_t are adjacent to x or y . In this case G therefore contains the graph $\Psi(i, j, k, r, s, t)$ in Figure 4.8 as spanning subgraph (where $i + j + k + r + s + t = n$).

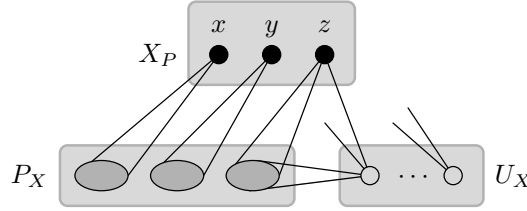


FIGURE 4.9: A spanning subgraph of G in Case 1 of the proof of Theorem 4.3.

Case 2: $x, y \in X_P$ and $z \in X_D$. In this case each vertex in D_X is adjacent to at least one vertex of X_P and each vertex in U_X is adjacent to both vertices of X_P in Figure 4.1, as depicted in Figure 4.10. Let \mathcal{U}_r be the set of $r \geq 0$ vertices that form a clique, \mathcal{K}_i (say), together with x and its private neighbours according to Theorem 3.6. Similarly, let \mathcal{V}_s be the set of $s \geq 0$ vertices that form a clique, \mathcal{K}_j (say), together with y and its private neighbours. For $t \geq 0$, let each vertex in $D_X = \mathcal{Z}_t$ be adjacent to z , where \mathcal{K}_k contains only the vertex z . Then each vertex of \mathcal{U}_r is adjacent to y , each vertex of \mathcal{V}_s is adjacent to x , each vertex of \mathcal{Z}_t is adjacent to z and to either x or y . In this case G therefore contains the graph $\Psi(i, j, 1, r, s, t)$ in Figure 4.8 as spanning subgraph (where $i + j + r + s + t + 1 = n$).

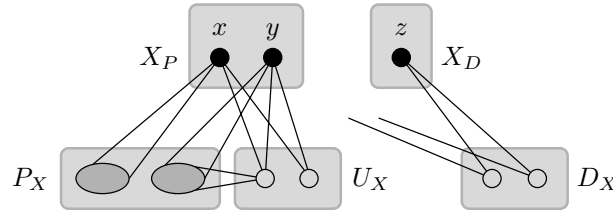


FIGURE 4.10: A spanning subgraph of G in Case 2 of the proof of Theorem 4.3.

Case 3: $x \in X_P$ and $y, z \in X_D$. In this case U_X is empty in Figure 4.1, for otherwise any vertex $u \in U_X$ would be a private neighbour of x , and each vertex in D_X is furthermore adjacent to at least one vertex in X_D , as depicted in Figure 4.11. Therefore, \mathcal{K}_j contains only the vertex y , \mathcal{K}_k contains only the vertex z , $D_X = \mathcal{V}_s \cup \mathcal{Z}_t$ and $\mathcal{U}_r = \emptyset$ in Figure 4.8 so that G contains the graph $\Psi(i, 1, 1, 0, s, t)$ in Figure 4.8 as spanning subgraph (where $i + s + t + 2 = n$).

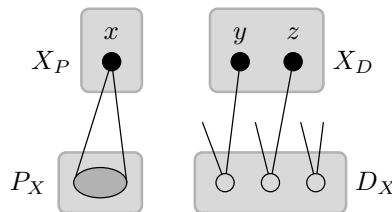
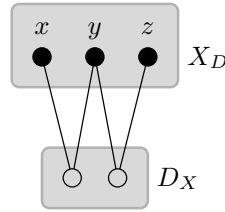
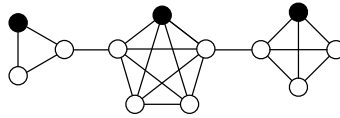


FIGURE 4.11: A spanning subgraph of G in Case 3 of the proof of Theorem 4.3.

Case 4: $x, y, z \in X_D$. In this case X_P , P_X and U_X are all empty in Figure 4.1 and each vertex in D_X is adjacent to at least two vertices in X_D , as depicted in Figure 4.12. Therefore, \mathcal{K}_i contains only the vertex x , \mathcal{K}_j contains only the vertex y , \mathcal{K}_k contains only the vertex z , and $D_X = \mathcal{U}_r \cup \mathcal{V}_s \cup \mathcal{Z}_t$ in Figure 4.8 so that G contains the graph $\Psi(1, 1, 1, r, s, t)$ in Figure 4.8 as spanning subgraph (where $i + s + t + 3 = n$). ■


 FIGURE 4.12: A spanning subgraph of G in Case 4 of the proof of Theorem 4.3.

The characterisation in Theorem 4.3 may be used to prove succinctly that $\gamma_s(G) \leq 3$ for a graph that contains $\Psi(i, j, k, r, s, t)$ in Figure 4.8 as spanning subgraph, by merely citing the particular spanning subgraph as certificate. Showing that $\gamma_s(G) \neq 2$ for such graphs may, however, prove more cumbersome.


 FIGURE 4.13: The double dumbbell graph $\mathcal{D}_{3,5,4}$.

Consider, as an example, the connected graph of order $a + b + c$ and size $\binom{a}{2} + \binom{b}{2} + \binom{c}{2} + 2$ obtained by joining three vertex disjoint cliques \mathcal{K}_a , \mathcal{K}_b and \mathcal{K}_c of orders $a \geq 2$, $b \geq 1$ and $c \geq 2$ in a chain by means of two additional edges, one edge joining any vertex in $V(\mathcal{K}_a)$ with any vertex in $V(\mathcal{K}_b)$, while the other edge joins any vertex in $V(\mathcal{K}_b)$ with any vertex in $V(\mathcal{K}_c)$, as illustrated in Figure 4.13. This graph is called the *double dumbbell graph* and is denoted by $\mathcal{D}_{a,b,c}$. Since the $\Psi(a - 1, b - 1, c, 1, 1, 0)$ is a spanning subgraph of the double dumbbell graph $\mathcal{D}_{a,b,c}$, it follows immediately from Theorem 4.3 that $\gamma_s(\mathcal{D}_{a,b,c}) \leq 3$. Further examples of certificates showing that $\gamma_s(G) \leq 3$ for various infinite classes of graphs G are shown in Table 4.2. Figure 4.14 contains graphical representations of four of the certificates listed in Table 4.2 in the context of the full graph classes in the table.

Graph class	Certificate
The complete bipartite graph less two edges, $\mathcal{K}_{2,n-2} - 2e$	$\Psi(2, 1, 1, 0, n - 4, 0)$
The complete tripartite graph $\mathcal{K}_{1,a,b}$ with $3 \leq a \leq b$	$\Psi(1, 1, 1, 0, 0, a + b - 2)$
The complete tripartite graph $\mathcal{K}_{a,b,c}$ with $3 \leq a \leq b \leq c$	$\Psi(1, 1, 1, b - 1, c - 1, a - 1)$
The cartesian product, $\mathcal{K}_3 \square \mathcal{K}_s$	$\Psi(s, s, s, 0, 0, 0)$
The double dumbbell graph, $\mathcal{D}_{a,b,c}$	$\Psi(a - 1, b - 1, c, 1, 1, 0)$

 TABLE 4.2: Certificates showing that $\gamma_s(G) \leq 3$ for various infinite classes of graphs G .

4.3 On the defenders in a minimum secure dominating set

It is interesting to note that not all members of a *minimum* secure dominating set of a graph need to be defenders, as illustrated in Figure 4.15.

It is, however, possible to increase the number of defenders successively in a minimum secure dominating set of any connected graph without end vertices, until all the members of the set are indeed defenders, as is shown next.

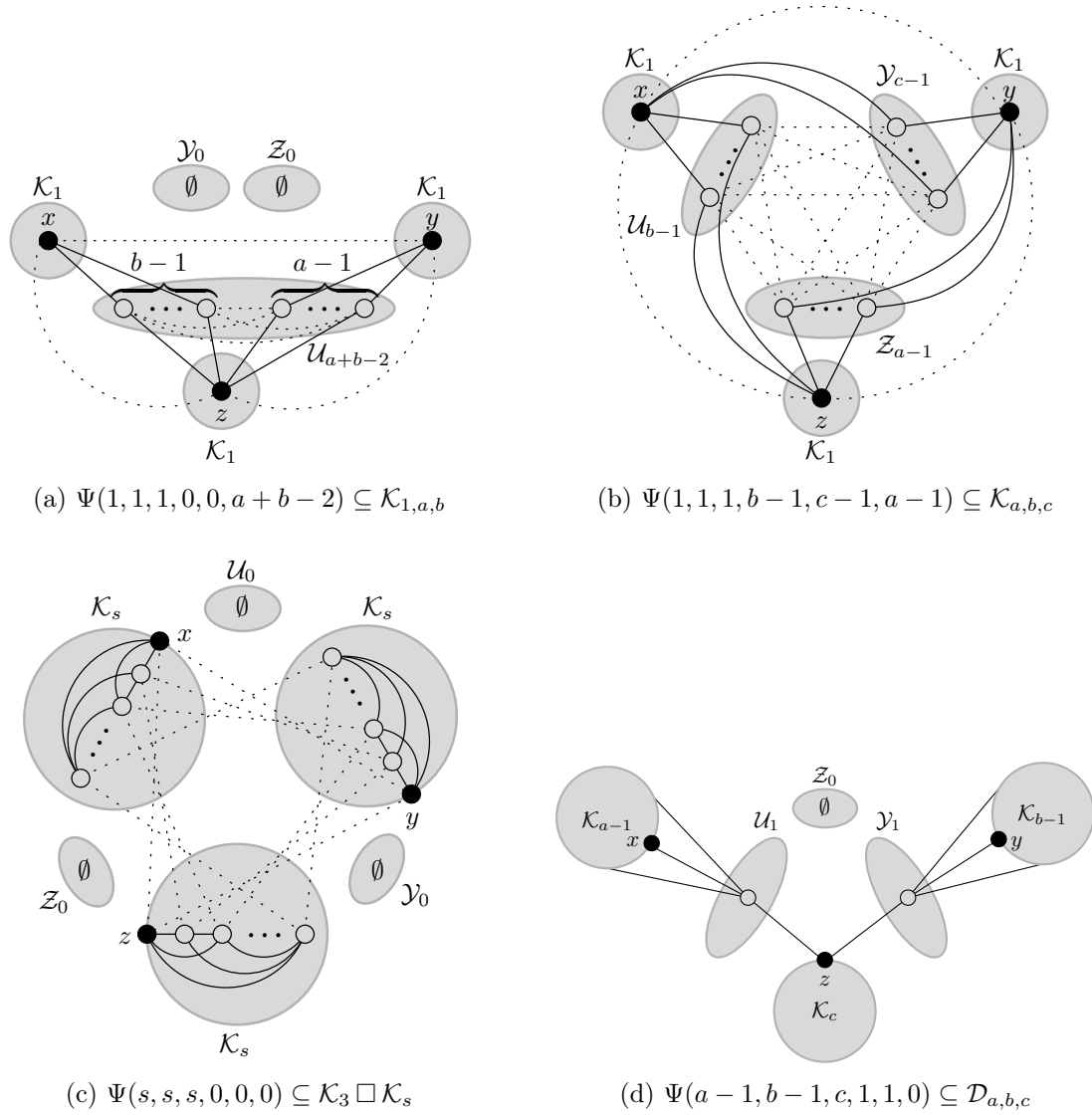


FIGURE 4.14: Various infinite graph classes together with certificates showing that the secure domination number of each of these graphs is at most 3. Dotted lines represent edges that are present in the graph classes of Table 4.2, but not in the certificates listed in the table. (a) The complete tripartite graph $\mathcal{K}_{1,a,b}$ with $3 \leq a \leq b$. (b) The complete tripartite graph $\mathcal{K}_{a,b,c}$ with $3 \leq a \leq b \leq c$. (c) The cartesian product $\mathcal{K}_3 \square \mathcal{K}_a$. (d) The double dumbbell graph $\mathcal{D}_{a,b,c}$.

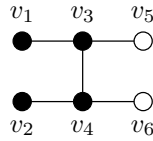


FIGURE 4.15: A graph of order 6 with a minimum secure dominating set denoted by solid vertices; v_3 defends v_5 , while v_4 defends v_6 . Although members of the minimum secure dominating set, the vertices v_1 and v_2 are not defenders.

Theorem 4.4 *If X is a minimum secure dominating set of a connected graph G without isolated vertices and some vertex in X is not a defender, then there exists another minimum secure dominating set of G which contains one more defender than X .*

Proof: Let G be a connected graph with minimum degree at least 2 and let X be a minimum secure dominating set of G in which some vertex $x \in X$ does not defend any vertex in $N(x)$. Then $\text{Epn}(x, X) = \emptyset$, for otherwise $\text{Epn}(x, X)$ would induce a clique in G , all of whose vertices are defended by x .

It is first shown, by contradiction, that $N(x) \subset X$. Suppose, to the contrary, that $N(x) \cap (V - X) \neq \emptyset$ and let v be a neighbour of x in $V - X$. Then v is adjacent to some vertex in $X - \{x\}$, and so the swap set $(X - \{x\}) \cup \{v\}$ is a dominating set of G . Therefore, x defends v , a contradiction.

It is next shown, again by contradiction, that $\text{Epn}(u, X) \neq \emptyset$ for all $u \in N(x)$. Suppose, to the contrary, that $\text{Epn}(u', X) = \emptyset$ for some vertex $u' \in N(x)$. Then each vertex in $N(u') \cap (V - X)$ is dominated by at least one vertex in $X - \{x, u'\}$ and u' defends all vertices in $N(u') \cap (V - X)$. But then $X - \{x\}$ is a secure dominating set of G , since each vertex in $V - X$ is defended by a vertex in X while x is defended by u' , contradicting the minimality of X .

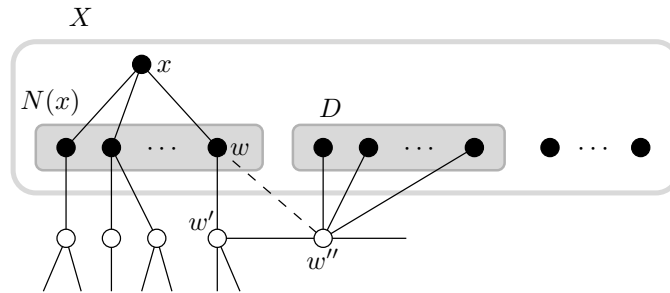


FIGURE 4.16: The secure dominating set in the proof of Theorem 4.4. The dashed connection between w and w'' may or may not be an element of the edge set of the graph.

Now let $w \in N(x)$ and suppose $w' \in \text{Epn}(w, X)$, as illustrated in Figure 4.16. If $\deg(w') = 1$, let $X^* = (X - \{w\}) \cup \{w'\}$. It follows that $\text{Epn}(w, X) = \{w'\}$ because $\text{Epn}(w, X) \cup \{w\}$ forms a complete graph. Furthermore, for the set X^* , the vertices x and w' defend w and the remaining vertices of $V - X^*$ are defended by the vertices that defended them in X .

Since the minimum degree of G is at least two, w' is adjacent to some vertex $w'' \in V(G) - X$. Let $D \subset X$ be the set of all vertices that defend w'' . Then $X' = (X - \{x\}) \cup \{w'\}$ is a secure dominating set of G in which w defends x , w' defends all the vertices in $N(w') \cap (V - X')$, w'' is defended by $D \cup \{w'\}$ and the remaining vertices of $V - X'$ are defended by the vertices that defended them in X . ■

The following result is an immediate consequence of a repeated application of Theorem 4.4, as illustrated in Figure 4.17.

Corollary 4.1 *If G is a connected graph without isolated vertices, then there exists a minimum secure dominating set of G in which every vertex is a defender.*

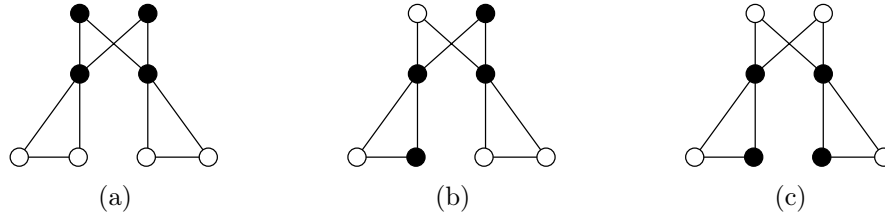


FIGURE 4.17: Minimum secure dominating sets of a graph of order 8 with minimum degree 2, containing (a) two, (b) three and (c) four defenders.

4.4 Secure domination is **NP**-complete

A polynomial-time reduction of the dominating set problem (as defined in Decision Problem 2.4) to the secure dominating set problem, formulated below, is demonstrated in this section.

Decision Problem 4.1 (Secure dominating set problem)

Instance: A graph $G = (V, E)$ and a positive integer $k \leq |V(G)|$.

Question: Does G have a secure dominating set of cardinality k or smaller?

The following proof is similar to the proof by Henning and Hedetniemi [62] that the weak Roman domination problem is **NP**-complete. Given a graph G , consider a graph H formed by adding to each vertex of G a pendent path of length 4. The following result is necessary in order to derive a polynomial-time reduction of the **NP**-complete Decision Problem 2.4 (see Theorem 2.36) to Decision Problem 4.1.

Lemma 4.1 $\gamma_s(H) = \gamma(G) + 2|V(G)|$.

Proof: Let $f = (V_0, V_1)$ be a secure dominating function of H , let $v \in V(G) \subset V(H)$, and let $\mathcal{P}_v : v, w, x, y, z$ be the pendent path of length 4 added to v during the construction of H from G , as shown in Figure 4.18(a). It follows that $f[u] \geq 1$ for every vertex u of H . In particular, $f[w] \geq 1$ and $f[z] \geq 1$, and so $f(V(\mathcal{P}_v)) = f[w] + f[z] \geq 2$. Assume that $f(z) = 0$ and $f(y) = 1$, and let $S = V_1 \cap V(G)$.

If $f(V(\mathcal{P}_v)) \geq 3$, then assume that $f(v) = f(w) = f(y) = 1$ and $f(x) = f(z) = 0$, as shown in Figure 4.18(b). Hence, if $f(V(\mathcal{P}_v)) \geq 3$, then $v \in S$. Suppose that $f(V(\mathcal{P}_v)) = 2$. Then $f[w] = 1$ and $f[z] = 1$. Thus, $f(z) = 0$ and $f(y) = 1$. If $f(x) = 1$, then $f(v) = f(w) = 0$. In particular, $v \in V_0$, and so v must be adjacent to a vertex $u \in V_1$, as shown in Figure 4.18(c). Since $w \in V_0$ ($f(w) = 0$), $u \in V(G)$. Hence, v is adjacent to a vertex in S . On the other hand, suppose $f(x) = 0$. Since the swap set $(V_1 - \{y\}) \cup \{x\}$ does not dominate z , it follows that $f(w) = 1$ and $f(v) = 0$, and that the swap set $(V_1 - \{w\}) \cup \{x\}$ does not dominate v . But this implies that the vertex v must be adjacent to a vertex $u \in S$, as shown in Figure 4.18(d). Hence, if $f(V(\mathcal{P}_v)) = 2$, then v is dominated by a vertex in S .

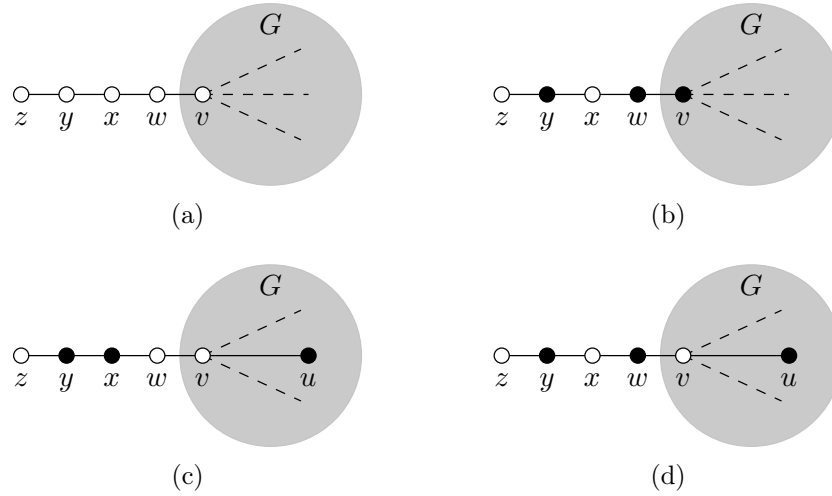


FIGURE 4.18: (a) The pendent path \mathcal{P}_v . (b) The case where $f(V(\mathcal{P}_v)) = 3$ such that $f(v) = f(w) = f(x) = 1$. (c) The case where $f(V(\mathcal{P}_v)) = 2$ such that $f(x) = f(y) = 1$. (d) The case where $f(V(\mathcal{P}_v)) = 2$ such that $f(w) = f(y) = 1$.

Thus, S is a dominating set of G , and so $\gamma(G) \leq |S|$. Furthermore, if $v \in S$, then $f(V(\mathcal{P}_v)) \geq 3$, while if $v \notin S$, then $f(V(\mathcal{P}_v)) = 2$. Hence,

$$\begin{aligned} \gamma_s(H) &= w(f) \\ &\geq 3|S| + 2(|V(G)| - |S|) \\ &= |S| + 2|V(G)| \\ &\geq \gamma(G) + 2|V(G)|. \end{aligned}$$

On the other hand, let D be a dominating set of G and let $g : V(H) \mapsto \{0, 1\}$ be the function defined as follows: if $v \in D$, then let $g(v) = g(w) = g(y) = 1$ and $g(x) = g(z) = 0$, while if $v \notin D$, then let $g(v) = g(x) = g(z) = 0$ and $g(w) = g(y) = 1$. Then g is a secure dominating function of H , and so

$$\begin{aligned} \gamma_s(H) &\leq w(g) \\ &= 3|D| + 2(|V(G)| - |D|) \\ &= |D| + 2|V(G)| \\ &= \gamma(G) + 2|V(G)|. \end{aligned}$$

Consequently, $\gamma_s(H) = \gamma(G) + 2|V(G)|$, as desired. ■

The following result is an immediate consequence of Lemma 4.1.

Theorem 4.5 *Decision Problem 4.1 is **NP**-complete.*

Proof: Let $D_{\text{SDS}}(G, j)$ denote the decision problem of, given a graph G of order n and a positive integer j , determining whether G contains a secure dominating set of cardinality $j \leq n$. Then $D_{\text{SDS}}(G, j)$ is in **NP**, a certificate being any secure dominating set of G of cardinality j . The construction of a graph H from G by adding a pendent path of length 4 to each vertex of G can clearly be accomplished in polynomial time. Lemma 4.1 implies that if $j = k + 2|V(G)|$, then $\gamma(G) \leq k$ if and only if $\gamma_s(H) \leq j$. Decision Problem 2.4 may therefore be reduced in polynomial time to Decision Problem 4.1. ■

4.5 Chapter summary

A number of basic results on the nature and computation of minimum secure dominating sets of arbitrary graphs were established in this chapter. The chapter opened in §4.1 with a description of three necessary and sufficient criteria for establishing whether or not a subset of the vertex set of a graph is, in fact, a secure dominating set of the graph. The vertex set of a graph G may be partitioned into five subsets with respect to any subset X of the vertex set of G , as demonstrated in Figure 4.1. The result of Theorem 3.6 was used to arrange the three above-mentioned criteria for secure graph domination in increasing order of computational complexity, resulting in an orderly and logical set of secure domination testing criteria.

The classes of graphs that have secure domination numbers 1, 2 or 3 were characterised in §4.2. The class of graphs with secure domination number 1 was characterised as being the class of complete graphs only. For the class of graphs with secure domination number 2, a graph construction was required, resulting in a certificate denoted by $\Phi(i, j, k, \ell)$. It was shown that any incomplete graph which contains $\Phi(i, j, k, \ell)$ as spanning subgraph for some integers $i, j \geq 1$ and $k, \ell \geq 0$ has a secure domination number of 2. Certificates were presented for showing that a number of infinite classes of graphs have secure domination number 2. A similar graph construction framework was presented for the class of graphs with secure domination number 3. In this case, however, a certificate denoted by $\Psi(i, j, k, r, s, t)$ is applicable for integers $i, j, k \geq 1$ and $r, s, t \geq 0$. Certificates were once again presented showing that a number of infinite classes of graphs admit secure dominating sets of cardinality 3.

It was shown in §4.3 that it is possible to successively increase the number of defenders in a minimum secure dominating set of a connected graph with minimum degree at least two, until all the members of the minimum secure dominating set are defenders.

The decision problem associated with secure domination was finally shown to be **NP-complete** in §4.4 by employing a proof technique similar to that of Henning and Hedetniemi [62], who showed that the decision problem associated with weak Roman dominating is **NP-complete**.

CHAPTER 5

Algorithms for secure domination

Contents

5.1	Computing the secure domination number of an arbitrary graph	79
5.1.1	A branch-and-reduce algorithm	81
5.1.2	A branch-and-bound algorithm	84
5.1.3	Worst-case complexity analysis	87
5.1.4	Numerical results	87
5.1.5	A binary programming problem formulation	90
5.2	Computing the secure domination number of a tree	92
5.2.1	Spiders	92
5.2.2	From spiders to trees in general	97
5.2.3	Linear algorithmic implementation	102
5.2.4	Validation of the algorithm	107
5.3	Chapter summary	107

This chapter contains a description of four algorithmic approaches towards finding the secure domination number of a graph. The chapter opens in §5.1 with two exact algorithms for finding the secure domination number of an arbitrary graph, namely a branch-and-reduce algorithm and a branch-and-bound algorithm. This is followed by a binary programming approach towards finding the secure domination number of an arbitrary graph. A linear algorithm is finally presented in §5.2 for finding the secure domination number of a tree.

5.1 Computing the secure domination number of an arbitrary graph

A set $R' \subseteq V(G)$ of vertices is called a *redundant set* of a graph G if any two distinct vertices $u, v \in R'$ share a common closed neighbourhood (i.e. $N[u] = N[v]$) which forms a clique in G . A maximal redundant set of G is called a *redundancy class* of G . It is shown that, without loss of generality, all support vertices of G may be included and all but one vertices of each redundancy class of G may be excluded when seeking a minimum secure dominating set of a graph G .

Theorem 5.1 *Let S be the set of all support vertices of a graph G , and let u and v be two distinct vertices in the same redundancy class of G . Then*

- (a) *there is a minimum secure dominating set X of G such that $S \subseteq X$.*
- (b) *no minimum secure dominating set of G contains both u and v .*

Proof: (a) Let X be a minimum secure dominating set of G that contains as many vertices in S as possible. If $S \not\subseteq X$, there is a support vertex s of G which is not in X . Therefore, all the end-vertices of G supported by s are necessarily included in X (because these end-vertices are private neighbours of s), and so $(X \cup \{s\}) - \{w\}$ is also a secure dominating set of G of cardinality $|X|$, where w is any one of the end-vertices of G supported by s .

(b) Suppose u, v are two distinct vertices in the same redundancy class of G and that these vertices form part of a secure dominating set X of G . Then $N[u] = N[v]$ and $G[N[u]] = G[N[v]]$ is complete, so that neither u nor v has private neighbours external to $\{u, v\}$. Therefore v defends the closed neighbourhood of u by Theorem 3.6, and hence $X - \{u\}$ is also a secure dominating set of G . ■

A connected subgraph of a tree induced by a support vertex and its adjacent leaves is called an *end-cluster* of the tree. The simple heuristic presented in pseudo-code form as Algorithm 5.1 may be used to find an upper bound on the secure domination number of a connected graph. The heuristic is based on the fact that an end-cluster of a tree is securely dominated by its leaves. Therefore, a secure dominating set X of a graph G may be obtained by computing a spanning tree of G and then including all the leaves of this tree in X , after which all end-clusters of the tree may be pruned away to form a smaller tree. The same pruning procedure may be applied to this smaller tree, after having inserted the newly formed leaves of this smaller tree into X , and so on, until all the vertices of the original spanning tree have been pruned away.

Algorithm 5.1: Heuristic

Input : A connected graph G .

Output: A secure dominating set of G .

```

1  $X \leftarrow \emptyset$ ;
2  $T \leftarrow$  A spanning tree of  $G$ ;
3 while  $V(T) \neq \emptyset$  do
4   | Insert all leaves of  $T$  into  $X$ ;
5   | Update  $T$  by removing all its end-clusters;
6 return  $[X]$ ;
```

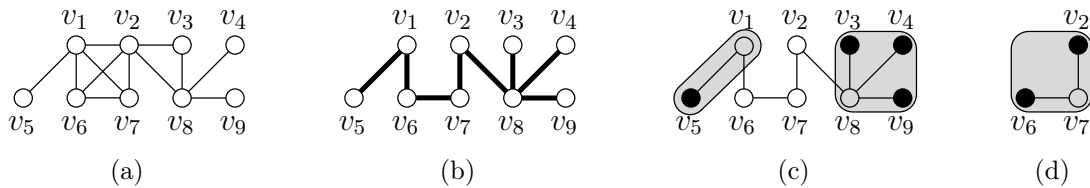


FIGURE 5.1: (a) A graph G_{10} for which $\gamma_s(G_{10}) = 4$. (b) A spanning tree of G_{10} . (c)–(d) The result of identifying and pruning away end-clusters of the spanning tree in (b) according to Algorithm 5.1 to arrive at the secure dominating set $\{v_2, v_3, v_4, v_5, v_6, v_9\}$ of cardinality 6 for G_{10} .

Consider, as an example, the graph G_{10} in Figure 5.1(a). A spanning tree of G_{10} is shown in Figure 5.1(b). During the first iteration of the while-loop spanning Steps 3–5 of Algorithm 5.1, the four leaves (indicated as solid vertices) in Figure 5.1(c) are inserted into the set X . Thereupon the two end-clusters (highlighted in grey in the figure) are removed from the tree to obtain the smaller tree in Figure 5.1(d). The two leaves of this smaller tree are also inserted into X after which the entire tree in the figure is pruned away (because the tree consists of a single end-cluster). This process results in the secure dominating set $X = \{v_2, v_3, v_4, v_5, v_6, v_9\}$ of cardinality 6 for the graph G_{10} .

5.1.1 A branch-and-reduce algorithm

The first (recursive) algorithm proposed in this chapter for computing the secure domination number of an arbitrary graph G is a so-called *branch-and-reduce algorithm*. This algorithm traverses a depth-first search tree to construct a secure dominating set X of G such that $X \subseteq V(G) - F$, where F denotes a set of specified, forbidden vertices (*i.e.* vertices that may not be included in X). The algorithm takes as input the quadruple (G, X_P, D_X, F) , where X_P and D_X are the sets described in §4.1. The sets P_X and U_X , also described in §4.1, may easily be computed from this quadruple. Initially, X_P is taken as the set S' of all support vertices of G while F is taken as the set R' of all but one vertices from each redundancy class of G , both without loss of generality, in view of Proposition 5.1, while the set D_X is initially empty. A standard branch-and-reduce approach is employed at each call of the algorithm in the sense that each vertex $v \in V(G) - S' - R'$ is considered in turn to be included either in X or F :

- Inclusion of a vertex v in X in the former case may annihilate external private neighbours of vertices already in X_P , resulting in a need to repartition the extended set $X' = X_P \cup X_D \cup \{v\}$ into a new subset X'_P of vertices with private neighbours external to X' and a subset X'_D with no such external private neighbours, in turn resulting in a new set D'_X of defended vertices. The graph may be reduced to $G - X'_D$, since the vertices in X'_D cannot annihilate external private neighbours of vertices in X'_P or any vertices that may be included in X'_P during later algorithmic calls. Furthermore, all the vertices in D'_X are defended. A smaller problem may then be solved by calling the algorithm recursively with the reduced quadruple $(G - X'_D, X'_P, D'_X, F)$ as input.
- In the latter case a smaller problem may be solved by simply calling the algorithm with input quadruple $(G, X_P, D_X, F \cup \{v\})$.

Algorithm 5.2: BR

Input : A graph G , the sets X_P and D_X (described in §4.1) and a forbidden set F .

Output: A minimum secure dominating set of G from among the vertices $V(G) - F$.

```

1 if  $X_P$  is a dominating set of  $G - D_X$  then
2   if for each  $s \in X_P$ ,  $G[\{s\} \cup \text{Epn}(s, X_P)]$  is complete then
3      $P_X \leftarrow \bigcup_{s \in X_P} \text{Epn}(s, X_P)$ ;  $U_X \leftarrow V(G) - X_P - P_X - D_X$ ;
4     if for each  $u \in U_X$  there exists a  $s \in X_P$  such that  $u$  is adjacent to all the vertices
5        $\text{Epn}(s, X_P)$  then
6         return  $\emptyset$ ;
7 if  $V(G) - X_P - F = \emptyset$  then disqualify branch;
8 Select any  $v \in V(G) - X_P - F$  with largest closed neighbourhood;
9 if  $N[v] = 1$  then return  $V(G) - X_P - F$ ;
10 else if  $N[v] = 2$  then return a single vertex from each component;
11  $X' \leftarrow \{v\} \cup X_P$ ;  $X'_P \leftarrow \bigcup_{x \in X'} \text{Epn}(x, X')$ ;  $X'_D \leftarrow X' - X'_P$ ;  $D'_X \leftarrow N(X'_D) \cap (V(G) - X')$ ;
return  $\min\{\{v\} \cup \text{BR}(G - X'_D, X'_P, D'_X, F), \text{BR}(G, X_P, D_X, F \cup \{v\})\}$ ;

```

The algorithm is presented in pseudo-code form as Algorithm 5.2. Initially, the algorithm tests whether X_P is a secure dominating set of $G - D_X$; this occurs in the three nested if-tests spanning Steps 1–5, which corresponds to testing the validity of requirements I–III in §4.1. If all three tests succeed (*i.e.* if X_P securely dominates $G - D_X$), a secure dominating set has been found and the algorithm returns an empty set in Step 5, thereby terminating the current branch of

the search tree. If no secure dominating set is found, the algorithm prepares for two further recursive calls to itself by selecting another branching vertex $v \in V(G) - X_P - F$ with largest closed neighbourhood to include in X (at this stage $X_D = \emptyset$). If all the vertices of G are either in X_P or in F (Step 6 of Algorithm 5.2), then no vertex can be considered as a branching vertex for which the branch of the search tree will yield a secure dominating set. Otherwise, two reduction rules are considered. The first reduction rule considers whether the closed neighbourhood of v is a singleton. If so, all the vertices in G are isolated, in which case G can only be securely dominated by including all the vertices in $V(G) - X_P - F$ in X , as performed in Step 8 of Algorithm 5.2.

If the closed neighbourhood of v is not a singleton, the second reduction rule is invoked, testing whether the closed neighbourhood of v is a 2-set. If so, the graph G consists of the vertex disjoint union of paths of orders 1 and 2. A secure dominating set of G may be formed by selecting a single vertex from each of its components, which occurs in Step 9 of Algorithm 5.2. If neither reduction rule applies, the branching rule is employed to generate two further, smaller problem instances when the algorithm calls itself recursively. The repartition of X' into two new sets X'_P and X'_D , as described above, occurs in Step 10, while the two branching calls to Algorithm 5.2 occur Step 11, which returns a minimum secure dominating set of G .

Algorithm 5.3: FindBR

Input : A connected graph G .

Output: Secure dominating set of minimum cardinality for G .

- 1 $S' \leftarrow$ set of support vertices of G ;
 - 2 $R' \leftarrow$ all but one vertices from each redundancy class of G ;
 - 3 **return** [BR(G, S', \emptyset, R')];
-

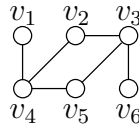


FIGURE 5.2: A graph G_{11} for which $\gamma_s(G_{11}) = 4$. G_{11} contains two support vertices, namely v_3 and v_4 , and no redundancy classes.

A search tree is constructed from the recursive calls to Algorithm 5.2. The root of this search tree is generated by the initial call $\mathbf{BR}(G, S', \emptyset, R')$ to Algorithm 5.2, which occurs in Step 3 of an external initialisation procedure for which a pseudo-code listing is presented in Algorithm 5.3. This latter algorithm takes as input a connected graph G and determines the set of support vertices S' and all but one vertices from each redundancy class R' of G . Algorithm 5.2 is called in Step 3 where S' and R' are used as input parameters to reduce the size of the search tree.

The search tree constructed by Algorithms 5.2 and 5.3 for the graph G_{11} in Figure 5.2 is shown in Figure 5.3. Since G_{11} has the set of support vertices $S' = \{v_3, v_4\}$, the set X is initialised as $\{v_3, v_4\}$ in the root of the tree, labelled node 0. Although X is a dominating set of G_{11} , it is not a secure dominating set of G_{11} . The vertices of $G_{11} - S'$ are branched in order of decreasing size of their respective closed neighbourhoods. In the case of a tie, vertices are selected in the order in which they are labelled. The tree is traversed in a depth-first fashion, and branching on the inclusion of v_2 into X_P or F produces the sets shown in nodes 1 and 8 of the search tree in Figure 5.3, respectively. The vertex v_2 is included in X in node 1 of the search tree for which the union of the sets $X_P = \{v_3, v_4\}$ and $X_D = \{v_2\}$ is not a secure dominating set of G_{11} . When branching on node 1, the vertex v_2 may be removed from G_{11} , since v_2 can defend itself

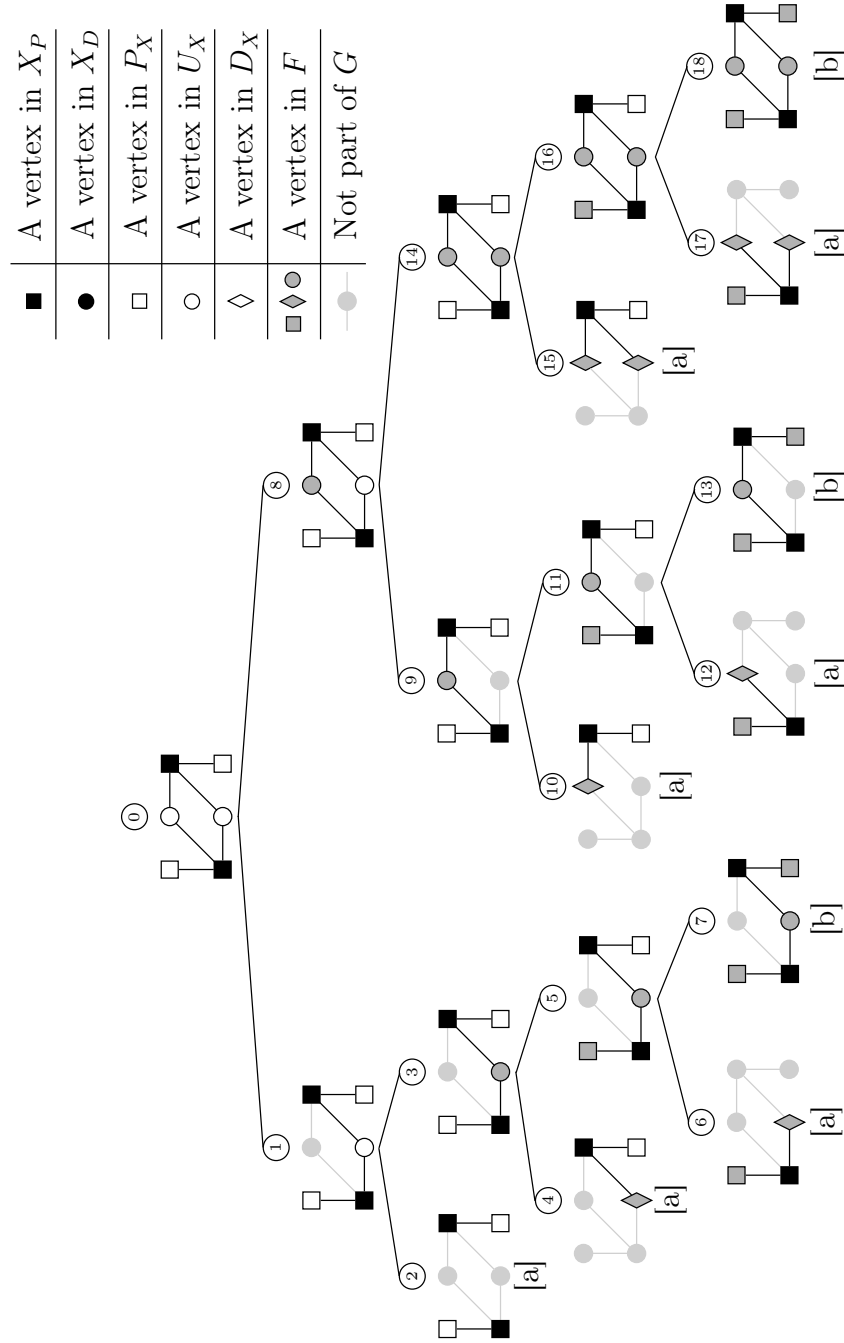


FIGURE 5.3: The secure dominating set depth-first search tree for the graph G_{11} in Figure 5.2 obtained by Algorithms 5.2 and 5.3. The order of traversing the tree is indicated by circled node numbers. Bounding the search tree is motivated as follows: [a] The tree is bounded in Step 5 of Algorithm 5.2, since a secure dominating set has been found. [b] The tree is bounded in Step 9 of Algorithm 5.2, because the vertices in X do not form a secure dominating set, while the remaining vertices are in F .

and cannot annihilate external private neighbours of any vertices that may later enter X_P . The vertex v_5 is chosen next to branch upon. This time the union of the sets $X_P = \{v_3, v_4\}$ and $X_D = \{v_2, v_5\}$ is a secure dominating set of cardinality 4 for G_{11} . The tree is therefore bounded in Step 5 of Algorithm 5.2, denoted by bounding reason “[a]” in the figure. Backtracking to node 3, the set of forbidden vertices is updated to $F = \{v_5\}$ and the vertex v_1 is chosen next to branch upon, leading to nodes 4 and 5 of the search tree. Node 4 is bounded since $X = \{v_1, v_2, v_3, v_4\}$

is a secure dominating set of cardinality 4 for G_{11} . In node 5 the set of forbidden vertices is updated to $F = \{v_1, v_5\}$ and the vertex v_6 is chosen next to branch upon in order to form nodes 6 and 7 of the search tree. Node 6 is bounded since $X = \{v_2, v_3, v_4, v_6\}$ is a secure dominating set of cardinality 4 for G_{11} . Node 7 is bounded since the vertices in $X = \{v_2, v_3, v_4\}$ do not form a secure dominating set of G_{11} and the remaining vertices of G are already in F ; hence a secure dominating set cannot be found. The tree is therefore bounded in Step 6 of Algorithm 5.2, denoted by the bounding reason “[b]” in the figure. Thereafter the search backtracks to node 8 and the same procedure is followed in the remaining ten nodes of the search tree. Minimum cardinality secure dominating sets of size 3 are found at node 15 with $X = \{v_1, v_3, v_4\}$ and node 17 with $X = \{v_3, v_4, v_6\}$.

5.1.2 A branch-and-bound algorithm

The second algorithm also constructs a secure dominating set X of G (recursively) from among the vertices in $V(G) - F$, where $F \subset V(G)$ again denotes a set of forbidden vertices, but this time the algorithm adopts a classical branch-and-bound approach. The algorithm takes as input the triple of sets (X_P, X_D, F) , where X_P and X_D are described in §4.1. From this triple, the sets P_X , U_X and D_X , also described in §4.1, may readily be determined. Initially X_P is again taken as the set S' of all support vertices of G and F is again taken as the set R' of all but one vertices from each redundancy class of G , without loss of generality in view of Proposition 5.1, while the set X_D is initially empty. A branching decision is taken with respect to each vertex v in $V(G) - S' - R'$ in turn, namely either to include it in $X = X_P \cup X_D$, or to include it in F :

- As in the branch-and-reduce approach of §5.1.1, the inclusion of a vertex v in X in the former case may annihilate external private neighbours of vertices already in X_P . The resulting set $X' = X_P \cup X_D \cup \{v\}$ must therefore again be repartitioned into a new subset X'_P of vertices with private neighbours external to X' and a subset X'_D with no such external private neighbours. A smaller problem may then be solved by calling the algorithm recursively with the triple (X'_P, X'_D, F) as input.
- In the latter case a smaller problem is solved by calling the algorithm with input triple $(X_P, X_D, F \cup \{v\})$.

The algorithm is presented in pseudo-code form as Algorithm 5.4. The difference between this algorithmic approach and the one in §5.1.1 is that three variables, **LowerBound**, **SmallestSetSoFar** and **UpperBound**, are defined and maintained globally (*i.e.* external to Algorithm 5.4), but Algorithm 5.4 is able to update the values of the latter two of these variables as smaller and smaller secure dominating sets are uncovered during the search. When the search is complete, the variable **SmallestSetSoFar** contains a minimum secure dominating set of cardinality **UpperBound** for the graph G . However, if at some point during execution of the algorithm, the set X has fewer than **LowerBound** elements, where **LowerBound** is a theoretical lower bound on $\gamma_s(G)$, then X cannot possibly be a secure dominating set of G and hence there is no need for testing the validity of any of the criteria I–III in §4.1. This observation is used to speed up the secure dominating set construction process in Step 1 of Algorithm 5.4. Furthermore, if so many vertices have been classified as forbidden (*i.e.* have been included in the set F and hence cannot be included in X) that $V(G) - F$ does not even dominate G , then no completion of X can be a secure dominating set of G . This is the reason for the bounding test in Step 2 of the algorithm. Finally, if the set X is so large that inclusion of one more vertex into the set will yield a set larger than **SmallestSetSoFar**, then the set X cannot be completed to a *smaller* secure dominating set than **SmallestSetSoFar**, thus giving rise to the bounding test in Step 3 of Algorithm 5.4.

Algorithm 5.4: BB

Input : The sets X_P and X_D in (4.1), and a set F of forbidden vertices.
Output: A minimum secure dominating set of G from amongst the vertices $V(G) - F$.

- 1 **if** $|X_P \cup X_D| < \text{LowerBound}$ *then* go to Step 12;
- 2 **else if** $V(G) - F$ does not dominate G **then return** $[]$;
- 3 **else if** $|X_P \cup X_D| \geq \text{UpperBound}$ **then return** $[]$;
- 4 $X \leftarrow X_P \cup X_D$;
- 5 **if** X is a dominating set of G **then**
- 6 **if** for each $s \in X_P$, $G[\{s\} \cup \text{Epn}(s, X)]$ is complete **then**
- 7 $P_X \leftarrow \bigcup_{s \in X_P} \text{Epn}(s, X)$; $D_X \leftarrow N(X_D) \cap (V(G) - X)$;
- 8 $U_X \leftarrow V(G) - X - D_X - P_X$;
- 9 **if** for each $u \in U_X$ there exists a $s \in X_P$ such that u is adjacent to all the vertices $\text{Epn}(s, X)$ **then**
- 10 $\text{UpperBound} \leftarrow |X|$;
- 11 $\text{SmallestSetSoFar} \leftarrow X$;
- 11 **return** $[]$;
- 12 Select any $v \in V(G) - X - F$;
- 13 $X' \leftarrow \{v\} \cup X_P \cup X_D$; $X'_P \leftarrow \bigcup_{x \in X'} \text{Epn}(x, X')$; $X'_D \leftarrow X' - X'_P$;
- 14 **BB**(X'_P, X'_D, F);
- 15 **BB**($X_P, X_D, F \cup v$);

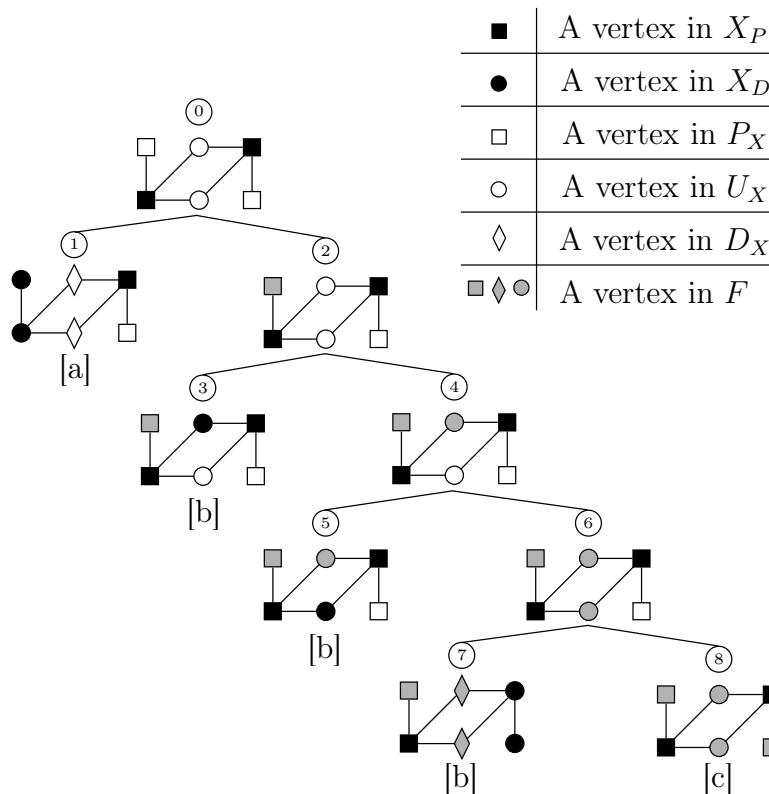
If all three of the above tests fail, then it must be determined whether X is a secure dominating set of G ; this occurs in the three nested if-tests spanning Steps 5–11 of the algorithm, which correspond to criteria I–III of §4.1, respectively. If all three of these tests succeed (*i.e.* if X securely dominates G), then a smaller secure dominating set than **SmallestSetSoFar** has been discovered, resulting in an update of the global variables **SmallestSetSoFar** and **UpperBound** in Steps 9 and 10 of Algorithm 5.4, after which the algorithm returns to the previous level of recursion in Step 11. If no such smaller secure dominating set has been found, the algorithm prepares for two further recursive calls to itself by selecting another branching vertex v to include in X so as to form the larger set $X' = X_P \cup X_D \cup \{v\}$ or alternatively to include it in F . The repartition of X' into two sets X'_P and X'_D , as described in §5.1.1, occurs in Step 13, while the two branching calls to Algorithm 5.4 occur in Steps 14 and 15.

Algorithm 5.5: FindBB

Input : A connected graph G of order n .
Output: A minimum secure dominating set of G .

- 1 $\text{LowerBound} \leftarrow \max \{ \lceil n/1 + \Delta \rceil, \lceil \text{diam}(G) + 1/3 \rceil \}$;
- 2 $\text{SmallestSetSoFar} \leftarrow$ A valid secure dominating set of G ;
- 3 $\text{UpperBound} \leftarrow |\text{SmallestSetSoFar}|$;
- 4 **if** $\text{LowerBound} = \text{UpperBound}$ **then return** $[\text{SmallestSetSoFar}]$;
- 5 **else**
- 6 $S' \leftarrow$ set of support vertices of G ;
- 7 $R' \leftarrow$ all but one vertices of each redundancy class of G ;
- 8 **BB**(S', \emptyset, R');
- 9 **return** $[\text{SmallestSetSoFar}]$;

A search tree may again be constructed from the recursive calls to Algorithm 5.4. The root of this search tree is generated by the initial call $\mathbf{BB}(S', \emptyset, R')$ to Algorithm 5.4, which occurs in Step 8 of an external initialisation procedure for which a pseudo-code listing is given in Algorithm 5.5. This latter algorithm takes as input a connected graph G of order n and determines the lower bound¹ LowerBound on $\gamma_s(G)$ in Step 1. Any valid (not necessarily minimum, but easily computable) secure dominating set SmallestSetSoFar of G is determined² in Step 2 of Algorithm 5.5, and the cardinality of this set is taken as the upper bound UpperBound on $\gamma_s(G)$ in Step 3 of the algorithm. If the bounds in Steps 1 and 3 coincide, then the secure dominating set computed in Step 2 is produced as output in Step 4; otherwise Algorithm 5.4 is called in Step 8.



$\{v_3, v_4\}$ in the root of the search tree, labelled node 0. Although X is a dominating set of G_{11} , it is not a secure dominating set of G_{11} . The vertices of $G_{11} - S'$ are branched upon in the order in which they are labelled. Branching on the inclusion of v_1 into X or F produces the sets shown in nodes 1 and 2 of the search tree in Figure 5.4, respectively. The tree is again traversed in a depth-first fashion, and the vertex v_1 is chosen next in node 1 of the search tree for which the set $X = \{v_1, v_3, v_4\}$ is a secure dominating set of cardinality 3 for G_{11} . The global variables **SmallestSetSoFar** and **UpperBound** are updated to reflect this discovery at node 1. The tree is therefore bounded in Step 11 of Algorithm 5.4, denoted by the bounding reason “[a]” in the figure. Backtracking to node 2, the set of forbidden vertices is updated from the empty set to $F = \{v_1\}$ and the vertex v_2 is chosen next to branch upon, leading to nodes 3 and 4 of the search tree. At node 3 the tree is immediately bounded in Step 3 of Algorithm 5.4, denoted by the bounding reason “[b]” in the figure, because at this node the set $X = \{v_2, v_3, v_4\}$ cannot be completed to a *smaller* secure dominating set of G_{11} than **SmallestSetSoFar**. Thereafter the search backtracks to node 4 and the same procedure is followed in the remaining four nodes of the search tree, never improving on **SmallestSetSoFar** = $\{v_1, v_3, v_4\}$ and finally terminating at node 8 where the tree is immediately bounded in Step 2 of Algorithm 5.4, denoted by the bounding reason “[c]” in the figure, because at this node the set $V(G) - F$ does not even dominate G_{11} . The set **SmallestSetSoFar** = $\{v_1, v_3, v_4\}$ is therefore smallest possible, leading to the result $\gamma_s(G_{11}) = \text{UpperBound} = 3$.

5.1.3 Worst-case complexity analysis

Since all but one vertices in a redundancy class and all support vertices may be included in X and F , respectively, before construction of the search trees starts, the worst-case time complexities of both algorithms in §5.1.1–§5.1.2 are $\mathcal{O}(2^{n-s-\sum_{i=1}^k(|R_i|-1)})$ for a graph of order n with s support vertices and R_1, \dots, R_k redundancy classes.

There is, however, a practical run-time trade-off between the number of recursive algorithmic calls and the time associated with each such call when one switches from the branch-and-bound paradigm to the branch-and-reduce paradigm. In particular, there are fewer algorithmic calls in the branch-and-bound algorithm of §5.1.2 than in the branch-and-reduce algorithm of §5.1.1 due to the incorporation of the global lower and upper bounds which are utilised to bound the search tree in the former case. However, whereas the input graph is globally defined and hence of the same order for all of the algorithmic calls in the former case, the input graph becomes smaller for recursive calls in the latter case as the search progresses deeper into the search tree. It is therefore not easy to make a general pronouncement as to which algorithm is expected to be faster in terms of actual computation time. For small graphs the classical branch-and-bound algorithm of §5.1.2 seems to be faster, as demonstrated in the next subsection.

5.1.4 Numerical results

Exact values of and bounds on the secure dominating numbers of members of specific graph classes were established by Cockayne *et al.* [32], as summarised in §3.2.2. In particular, general bounds were presented for grid graphs in the plane ($\mathcal{P}_m \square \mathcal{P}_k$) and for grid graphs on the torus ($\mathcal{C}_m \square \mathcal{C}_k$). Some numerical results obtained via the algorithms in §5.1.1 and §5.1.2 for these graph classes are presented in Table 5.1.

In the table, the column labelled Calls contains the number of times each algorithm calls itself recursively. All times are measured in seconds. It has been verified that the values of γ_s determined via the algorithms in §5.1.1 and §5.1.2 are indeed within the bounds established by

	Graph	n	γ_s	Branch-and-reduce (§5.1.1)		Branch-and-bound (§5.1.2)			
				Calls	Time	LB	UB	Calls	Time
Grids in the plane	$\mathcal{P}_2 \square \mathcal{P}_2$	4	2	21	0.01	2	2	21	0.01
	$\mathcal{P}_2 \square \mathcal{P}_3$	6	3	87	0.04	2	3	89	0.04
	$\mathcal{P}_2 \square \mathcal{P}_4$	8	4	359	0.17	2	4	251	0.15
	$\mathcal{P}_2 \square \mathcal{P}_5$	10	4	1 593	0.85	3	4	819	0.64
	$\mathcal{P}_2 \square \mathcal{P}_6$	12	5	6 479	3.89	3	5	2 049	2.26
	$\mathcal{P}_2 \square \mathcal{P}_7$	14	6	26 111	17.50	4	6	7 539	11.63
	$\mathcal{P}_2 \square \mathcal{P}_8$	16	7	107 467	79.38	4	7	19 867	43.61
	$\mathcal{P}_2 \square \mathcal{P}_9$	18	7	438 805	353.69	5	7	58 849	153.63
	$\mathcal{P}_2 \square \mathcal{P}_{10}$	20	8	1 783 065	1 570.81	5	8	141 673	501.85
	$\mathcal{P}_2 \square \mathcal{P}_{11}$	22	9	7 250 823	7 015.41	6	9	507 499	2 147.22
	$\mathcal{P}_2 \square \mathcal{P}_{12}$	24	10	29 443 157	30 764.65	6	10	1 291 811	7 437.22
	$\mathcal{P}_3 \square \mathcal{P}_3$	9	4	695	0.37	2	4	383	0.27
	$\mathcal{P}_3 \square \mathcal{P}_4$	12	5	5 971	3.74	3	5	1 939	2.05
	$\mathcal{P}_3 \square \mathcal{P}_5$	15	6	49 471	35.53	3	6	8 987	17.88
	$\mathcal{P}_3 \square \mathcal{P}_6$	18	7	414 835	341.40	4	7	39 279	109.07
	$\mathcal{P}_3 \square \mathcal{P}_7$	21	8	3 420 847	3 164.34	5	8	267 851	1 154.19
	$\mathcal{P}_3 \square \mathcal{P}_8$	24	10	28 175 363	29 321.29	5	10	1 209 227	7 990.45
	$\mathcal{P}_4 \square \mathcal{P}_4$	16	7	101 207	76.93	4	7	20 933	48.61
	$\mathcal{P}_4 \square \mathcal{P}_5$	20	8	1 684 617	1 525.13	4	8	118 315	451.01
	$\mathcal{P}_4 \square \mathcal{P}_6$	24	9	27 889 381	29 366.20	5	9	848 199	4 989.06
	$\mathcal{P}_5 \square \mathcal{P}_5$	25	9	56 630 783	61 212.33	5	9	1 511 009	10 101.53
Grids on the torus	$\mathcal{C}_2 \square \mathcal{C}_3$	6	2	57	0.03	2	2	43	0.01
	$\mathcal{C}_2 \square \mathcal{C}_4$	8	4	343	0.18	2	4	289	0.19
	$\mathcal{C}_2 \square \mathcal{C}_5$	10	4	1 373	0.81	3	4	717	0.53
	$\mathcal{C}_2 \square \mathcal{C}_6$	12	5	5 805	3.91	3	5	2 497	3.16
	$\mathcal{C}_2 \square \mathcal{C}_7$	14	6	23 817	17.57	4	6	7 947	12.25
	$\mathcal{C}_2 \square \mathcal{C}_8$	16	6	98 325	78.79	4	6	13 475	23.24
	$\mathcal{C}_2 \square \mathcal{C}_9$	18	7	405 101	352.78	5	7	58 979	147.63
	$\mathcal{C}_2 \square \mathcal{C}_{10}$	20	8	1 665 483	1 577.66	5	8	141 643	468.39
	$\mathcal{C}_2 \square \mathcal{C}_{11}$	22	9	6 833 333	7 020.25	6	9	551 461	2 294.11
	$\mathcal{C}_2 \square \mathcal{C}_{12}$	24	10	27 954 993	31 173.99	6	10	1 531 511	9 057.62
	$\mathcal{C}_3 \square \mathcal{C}_3$	9	3	417	0.25	2	3	231	0.11
	$\mathcal{C}_3 \square \mathcal{C}_4$	12	4	4 131	2.70	3	4	1 687	1.56
	$\mathcal{C}_3 \square \mathcal{C}_5$	15	5	33 189	25.17	3	5	7 275	11.10
	$\mathcal{C}_3 \square \mathcal{C}_6$	18	6	277 793	239.87	4	6	36 361	81.38
	$\mathcal{C}_3 \square \mathcal{C}_7$	21	7	2 334 103	2 238.20	5	7	184 881	573.56
	$\mathcal{C}_3 \square \mathcal{C}_8$	24	8	19 536 193	20 724.33	5	8	719 527	3 326.37
	$\mathcal{C}_4 \square \mathcal{C}_4$	16	6	81 127	68.51	4	6	19 121	35.87
	$\mathcal{C}_4 \square \mathcal{C}_5$	20	7	1 370 737	1 390.53	4	7	114 751	362.28
	$\mathcal{C}_4 \square \mathcal{C}_6$	24	8	23 233 767	26 887.09	5	8	715 993	3 349.03
	$\mathcal{C}_5 \square \mathcal{C}_5$	25	9	46 105 853	52 294.60	5	9	1 940 223	11 167.51

TABLE 5.1: Results obtained by the algorithms in §5.1.1 and §5.1.2 for all grid graphs in the plane and grid graphs on the torus of orders not exceeding $n = 25$. The columns labelled LB and UB contain the initial values of the global variables LowerBound and UpperBound (as described in §5.1.2) for the branch-and-bound algorithm.

Graph	n	$ R' $	γ_s	BR (§5.1.1)		BB (§5.1.2)			
				Calls	Time	LB	UB	Calls	Time
$C_3\langle 1 \rangle$	3	1	1	3	0.01	1	1	3	0.01
$C_4\langle 1 \rangle$	4	0	2	21	0.01	2	2	21	0.01
$C_4\langle 1, 2 \rangle$	4	1	1	3	0.01	1	2	3	0.01
$C_5\langle 1 \rangle$	5	0	3	51	0.03	2	3	51	0.02
$C_5\langle 1, 2 \rangle$	5	1	1	3	0.01	1	1	3	0.01
$C_6\langle 1 \rangle$	6	0	3	95	0.04	2	3	83	0.04
$C_6\langle 1, 2 \rangle$	6	0	2	43	0.02	2	2	43	0.02
$C_6\langle 1, 3 \rangle$	6	0	3	85	0.04	2	3	83	0.04
$C_6\langle 2, 3 \rangle$	6	0	2	57	0.03	2	2	43	0.02
$C_6\langle 1, 2, 3 \rangle$	6	1	1	3	0.01	1	1	3	0.01
$C_7\langle 1 \rangle$	7	0	3	199	0.09	3	3	149	0.07
$C_7\langle 1, 2 \rangle$	7	0	2	109	0.07	2	3	75	0.03
$C_7\langle 1, 2, 3 \rangle$	7	1	1	3	0.01	1	1	3	0.01
$C_8\langle 1 \rangle$	8	0	4	411	0.21	3	4	285	0.17
$C_8\langle 1, 2 \rangle$	8	0	3	217	0.14	2	3	189	0.11
$C_8\langle 1, 3 \rangle$	8	0	4	345	0.16	2	4	319	0.24
$C_8\langle 1, 4 \rangle$	8	0	3	317	0.18	2	3	199	0.11
$C_8\langle 1, 2, 3 \rangle$	8	0	2	73	0.05	2	2	73	0.03
$C_8\langle 1, 2, 4 \rangle$	8	0	2	117	0.08	2	3	73	0.03
$C_8\langle 1, 3, 4 \rangle$	8	0	2	87	0.06	2	2	73	0.03
$C_8\langle 1, 2, 3, 4 \rangle$	8	1	1	3	0.00	1	1	3	0.02
$C_9\langle 1 \rangle$	9	0	4	851	0.47	3	4	459	0.29
$C_9\langle 1, 2 \rangle$	9	0	3	367	0.21	2	3	265	0.15
$C_9\langle 1, 3 \rangle$	9	0	3	465	0.26	2	3	233	0.11
$C_9\langle 1, 2, 3 \rangle$	9	0	2	191	0.15	2	3	117	0.06
$C_9\langle 1, 2, 4 \rangle$	9	0	3	291	0.15	2	3	259	0.17
$C_9\langle 1, 2, 3, 4 \rangle$	9	1	1	3	0.01	1	1	3	0.01
$C_{10}\langle 1 \rangle$	10	0	5	1 731	1.00	4	5	1 133	0.91
$C_{10}\langle 1, 2 \rangle$	10	0	3	757	0.45	2	3	385	0.24
$C_{10}\langle 1, 3 \rangle$	10	0	4	1 233	0.76	2	4	687	0.54
$C_{10}\langle 1, 4 \rangle$	10	0	4	1 169	0.63	2	4	673	0.55
$C_{10}\langle 1, 5 \rangle$	10	0	4	1 343	0.81	3	4	715	0.51
$C_{10}\langle 2, 5 \rangle$	10	0	4	1 323	0.78	3	4	727	0.54
$C_{10}\langle 1, 2, 3 \rangle$	10	0	2	413	0.31	2	3	175	0.10
$C_{10}\langle 1, 2, 4 \rangle$	10	0	2	249	0.17	2	3	111	0.05
$C_{10}\langle 1, 2, 5 \rangle$	10	0	3	779	0.46	2	3	399	0.26
$C_{10}\langle 1, 3, 5 \rangle$	10	0	4	1 253	0.64	2	4	751	0.67
$C_{10}\langle 1, 4, 5 \rangle$	10	0	3	409	0.31	2	3	337	0.25
$C_{10}\langle 2, 4, 5 \rangle$	10	0	2	289	0.18	2	2	111	0.04
$C_{10}\langle 1, 2, 3, 4 \rangle$	10	0	2	111	0.10	2	2	111	0.05
$C_{10}\langle 1, 2, 3, 5 \rangle$	10	0	2	211	0.18	2	3	111	0.05
$C_{10}\langle 1, 2, 4, 5 \rangle$	10	0	2	211	0.19	2	3	111	0.05
$C_{10}\langle 1, 2, 3, 4, 5 \rangle$	10	1	1	3	0.01	1	1	3	0.01
$C_{11}\langle 1 \rangle$	11	0	5	3 529	2.14	4	5	1 627	1.35
$C_{11}\langle 1, 2 \rangle$	11	0	3	1 537	1.01	3	3	669	0.47
$C_{11}\langle 1, 3 \rangle$	11	0	4	2 301	1.45	3	4	1 107	0.93
$C_{11}\langle 1, 2, 3 \rangle$	11	0	3	711	0.53	2	3	481	0.35
$C_{11}\langle 1, 2, 4 \rangle$	11	0	3	1 129	0.70	2	3	467	0.30
$C_{11}\langle 1, 2, 3, 4 \rangle$	11	0	2	297	0.27	2	3	167	0.09
$C_{11}\langle 1, 2, 3, 4, 5 \rangle$	11	1	1	3	0.01	1	1	3	0.02

TABLE 5.2: Results obtained by the algorithms in §5.1.1 (BR) and §5.1.2 (BB) for all non-isomorphic, connected circulants of orders not exceeding $n = 11$. The columns labelled LB and UB contain the initial values of the global variables LowerBound and UpperBound (as described in §5.1.2) for the branch-and-bound algorithm. The column labelled $|R'|$ contains the number of redundancy classes in each graph.

Cockayne *et al.* [32]. No support vertices or redundant vertices are present in the examples of Table 5.1. Both the algorithms were coded in **Mathematica 8.0** [106] and executed on an Intel(R) Core(TM)2 Duo 3GHz processor with 3.7 GB RAM running on **Linux Ubuntu** [96].

Similar results are presented in Table 5.2 for all non-isomorphic, connected circulants of orders $n \leq 11$.

As a simple verification mechanism it was confirmed that both algorithms produced the same value of γ_s for all the above-mentioned test graphs.

Since the numerical results reported in this section were obtained via computer implementations in the high-level programming language **Mathematica 8.0** [106], the run-times in Tables 5.1–5.2 are relatively long. Experience has shown that speed-ups of an order of magnitude of 10^2 are typically possible with implementations in low-level languages such as C or C++.

5.1.5 A binary programming problem formulation

Let G be a graph of order n with vertex set $V(G) = \{v_1, \dots, v_n\}$. Furthermore, suppose the entry in row i and column j of the adjacency matrix of G is denoted by a_{ij} for all $i \neq j$, with the special convention that $a_{ii} = 1$ for all $i = 1, \dots, n$. Suppose a minimum secure dominating set X as sought for G and define the binary decision variables

$$x_i = \begin{cases} 1 & \text{if } v_i \in X \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

for all $i = 1, \dots, n$. Define, in addition to the binary decision variables in (5.1), the auxiliary binary variables

$$y_{k\ell} = \begin{cases} 1 & \text{if vertices } v_k \text{ } (x_k = 0) \text{ and } v_\ell \text{ } (x_\ell = 1) \text{ form a swap set} \\ 0 & \text{otherwise} \end{cases}$$

for all $k, \ell = 1, \dots, n$ and $k \neq \ell$.

The problem of computing a secure dominating set of minimum cardinality for G may be formulated as a binary program in which the objective is to

$$\text{minimise } z_s = \sum_{i=1}^n x_i \quad (5.2)$$

subject to the constraints

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, \dots, n, \quad (5.3)$$

$$\sum_{\substack{\ell=1 \\ \ell \neq k}}^n y_{k\ell} \geq 1 - x_k, \quad k = 1, \dots, n, \quad (5.4)$$

$$a_{k\ell}(x_\ell - x_k + 1) \geq 2y_{k\ell}, \quad \begin{matrix} k = 1, \dots, n, \\ \ell = 1, \dots, n, \quad \ell \neq k, \end{matrix} \quad (5.5)$$

$$a_{ik} + \sum_{\substack{j=1 \\ j \neq k \\ j \neq \ell}}^n a_{ij} x_j \geq y_{k\ell}, \quad \begin{matrix} i = 1, \dots, n, \\ k = 1, \dots, n, \\ \ell = 1, \dots, n, \quad \ell \neq k. \end{matrix} \quad (5.6)$$

Constraint set (5.3) ensures that each vertex v_i is adjacent to at least one vertex in X (*i.e.* that X is a dominating set of G). Constraint set (5.4) ensures, if $v_k \notin X$ (*i.e.* $x_k = 0$), that there is a swap set involving v_k , while constraint set (5.5) ensures that each swap set from v_ℓ to v_k ($y_{k\ell} = 1$) is valid (*i.e.* that $v_k \notin X$ and $v_\ell \in X$). Finally, constraint set (5.6) ensures that the configuration remains dominating after any single swap from v_ℓ to v_k is performed. This binary programming formulation consists of $n(n+1)$ variables and $n(n^2+n+2)$ constraints.

The binary program (5.2)–(5.6) was solved for the families of square grid graphs in the plane and square hexagonal graphs using CPLEX [37]. These graph classes were chosen as test instances because of their frequent use in war games [84, p. 116] and geographic information system applications [23, p. 23]. The results thus obtained for the $5 \square 5$ grid graph $\mathcal{P}_5 \square \mathcal{P}_5$ and the $5 \square 5$ hexagonal graph $\mathcal{H}_{5,5}$ are shown as an example in Figure 5.5. The results obtained for grid graphs and hexagonal graphs of other orders are summarised in Table 5.3. All numerical results were computed on an Intel(R) Core(TM)i7-3770 CPU 3.40GHz processor with 8.0 Gb RAM running in Linux Ubuntu [96].

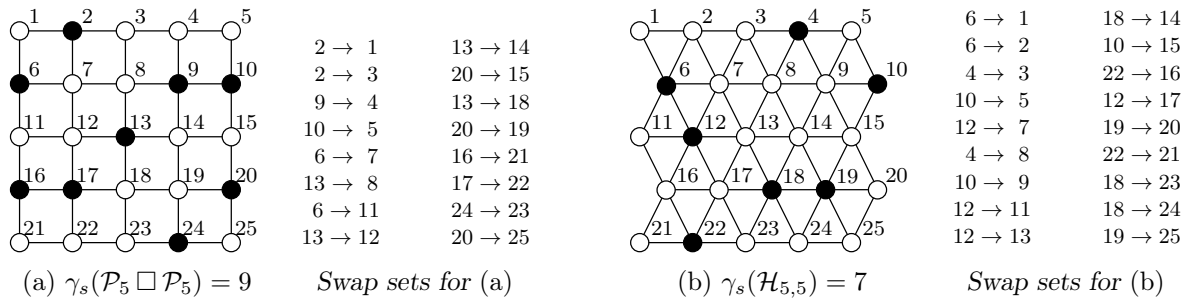


FIGURE 5.5: (a) An example of a secure dominating set of the square grid graph of order 25 in the plane. (b) An example of a secure dominating set of the square hexagonal graph of order 25. The swap sets corresponding to the secure dominating sets in (a) and (b) are also shown.

Results for $\mathcal{P}_n \square \mathcal{P}_n$				Results for $\mathcal{H}_{n,n}$			
n	LP	γ_s	Time	n	LP	γ_s	Time
2	1.33	2	0.01	2	1.00	2	0.01
3	2.50	4	0.02	3	2.00	3	0.01
4	4.00	7	0.07	4	4.00	5	0.04
5	6.27	9	0.17	5	5.00	7	0.08
6	8.75	13	0.39	6	6.00	10	0.27
7	11.50	18	2.90	7	9.00	13	7.41
8	14.81	23	72.16	8	11.33	17	110.00
9	18.25	29	23 356.24	9	13.50	21	1 083.17
10	22.39	35*	TO	10	16.86	27†	MO

TABLE 5.3: Results obtained by solving the binary program (5.2)–(5.6) for square grid graphs in the plane ($\mathcal{P}_n \square \mathcal{P}_n$) and “square” hexagonal graphs ($\mathcal{H}_{n,n}$). The values shown in columns labelled ‘LP’ are the associated linear programming relaxation lower bounds. All times are measured in seconds. A time-out bound of 8 hours (28 800 seconds) was enforced. If a problem instance required more than 8 hours of computation time, the acronym TO is shown in the time column and the time-out upper bound value on the parameter is accompanied by an asterisk. A memory-out bound of 8 Gb was enforced. If a problem instance required more than 8 Gb of memory, the acronym MO is shown in the time column and the memory-out upper bound value on the parameter is accompanied by a dagger.

5.2 Computing the secure domination number of a tree

Any vertex of degree at least 3 in a tree T is called a *branch vertex* of T . If T contains no branch vertex, then it is a path. An *endpath* \mathcal{P} of a tree T is a subpath of T that contains a leaf ℓ of T and in which every vertex $v \neq \ell$ has degree 2 in T . The following result is central to the development of the algorithm for secure domination of trees presented in this section.

Lemma 5.1 *Let T be a tree and let \mathcal{P} be an endpath of order j in T . Then there is no secure dominating set of T containing fewer than $\lceil 3(j-1)/7 \rceil$ vertices of \mathcal{P} .*

Proof: For $1 \leq j \leq 3$, the quantity $\lceil 3(j-1)/7 \rceil - 1$ is non-positive. For $j = 4$, there is only one possible dominating set of cardinality 1 for \mathcal{P} , and then only in the best-case scenario where the vertex immediately outside \mathcal{P} in T is included in the dominating set, as shown in Figure 5.6(a). For $j = 5$, there is not even a dominating set of cardinality 1 for \mathcal{P} , let alone a secure dominating set. For $j = 6$, there are three possible dominating sets of cardinality 2 for \mathcal{P} (again in the best-case scenario where the vertex immediately outside \mathcal{P} in T is included in the dominating set), as shown in Figure 5.6(b). Since none of these dominating sets is a secure dominating set of \mathcal{P} , the statement is therefore true for $j \leq 6$.

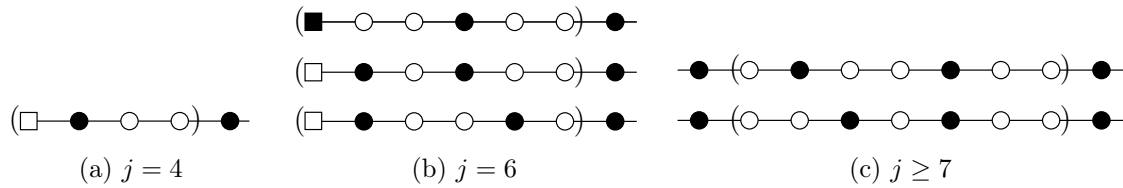


FIGURE 5.6: Solid vertices form part of the dominating sets of cardinality $\lceil 3(j-1)/7 \rceil - 1$ for the subpaths in brackets. Leaves are denoted by square vertices. None of these dominating sets is a secure dominating set.

Furthermore, any stretch of seven consecutive vertices in \mathcal{P} requires at least three vertices in any secure dominating set of \mathcal{P} for $j \geq 7$. This may be seen by noting that neither of the only two dominating sets of cardinality 2 for the subpath of order 7 in Figure 5.6(c) securely dominates the subpath, even in the best-case scenario where the two vertices immediately outside the subpath are both included in the dominating set.

Now suppose $j = 7s + t$ for some integer $0 \leq t < 7$. Then it follows that at least

$$\begin{aligned} \left\lceil \frac{3(t-1)}{7} \right\rceil + 3s &> \left\lceil \frac{3(t-1)}{7} \right\rceil - 1 + 3s \\ &= \left\lceil \frac{3(7s+t-1)}{7} \right\rceil - 1 \\ &= \left\lceil \frac{3(j-1)}{7} \right\rceil - 1 \end{aligned}$$

vertices are required to dominate \mathcal{P} securely. ■

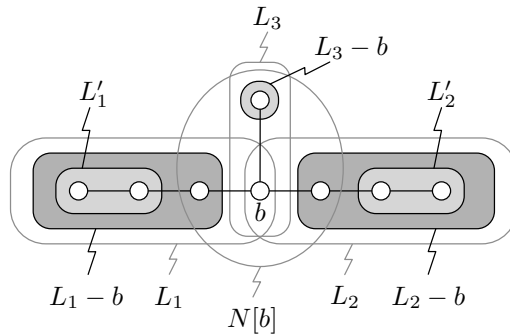
5.2.1 Spiders

The notion of a spider³ plays an important role in the algorithmic approach towards determining a minimum secure dominating set of an arbitrary tree. Recall, from §2.1.4, that a *spider* $\mathcal{S} =$

³This special kind of tree is sometimes also called a *wounded spider*.

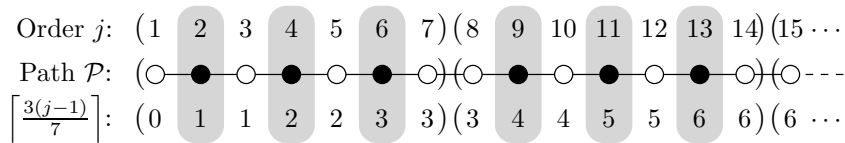
$\mathcal{S}(a_1, \dots, a_r)$ is a tree formed by joining $r \geq 1$ vertex-disjoint paths of orders a_1, \dots, a_r as pendent paths to a single vertex b , which is called the *anchor* of \mathcal{S} . Note, therefore, that a path is also a spider (*i.e.* the cases where $r = 1$ or $r = 2$). Furthermore, the only vertex of a spider which can be a branch vertex is its anchor, *i.e.* when $r \geq 3$ (but this need not be the case).

Let L_i be the path between the i -th leaf z_i of a spider \mathcal{S} and its anchor b . The path $L_i - b$ is called the i -th *leg* of \mathcal{S} , the reduced path $L'_i = L_i - N[b]$ is called its i -th *reduced leg* and the full path L_i the i -th *extended leg* of the spider. These notions are illustrated in Figure 5.7 for the spider $\mathcal{S}(3, 3, 1)$. Note that all legs and reduced legs of a spider are therefore also endpaths of the spider. The extended legs of a spider with at least three legs are, however, not endpaths of the spider.


 FIGURE 5.7: The spider $\mathcal{S}(3, 3, 1)$ with anchor b .

A *characteristic vector* (a_1^*, \dots, a_r^*) is associated with a spider $\mathcal{S}(a_1, \dots, a_r)$ where a_i^* is the residue of a_i after division by 7 (that is, $a_i^* \equiv a_i \pmod{7}$) for all $i = 1, \dots, r$. Define $\Lambda_i(\mathcal{S})$ as the unique subset of the vertex set of the i -th extended leg L_i of \mathcal{S} with the property that the vertex set of *every* endpath of order j within L_i shares exactly $\lceil 3(j-1)/7 \rceil$ vertices with $\Lambda_i(\mathcal{S})$, for all $1 \leq i \leq r$. Then the set $\Lambda_i(\mathcal{S})$ contains the solid vertices shown in Figures 5.8 and 5.9. Furthermore, let

$$\Lambda(\mathcal{S}) = \bigcup_{i=1}^r \Lambda_i(\mathcal{S}). \quad (5.7)$$


 FIGURE 5.8: The unique subset $\Lambda_i(\mathcal{S})$ of the vertex set of an extended leg L_i of a spider \mathcal{S} with the property that the vertex set of every endpath of L_i of order j shares exactly $\lceil 3(j-1)/7 \rceil$ vertices with $\Lambda_i(\mathcal{S})$.

The following result is an immediate consequence of Lemma 5.1.

Corollary 5.1 $\gamma_s(\mathcal{S}) \geq |\Lambda(\mathcal{S})|$ for any spider \mathcal{S} .

Proof: By contradiction. Suppose there exists a secure dominating set X of cardinality $|\Lambda(\mathcal{S})| - 1$ for the spider $\mathcal{S} = \mathcal{S}(a_1, \dots, a_r)$ with anchor b . Then either $b \in \Lambda(\mathcal{S})$ but $b \notin X$, or else X

contains fewer than $|\Lambda_i(\mathcal{S}) \cap V(L_i - b)|$ vertices from the leg $L_i - b$, for some $i \in \{1, \dots, r\}$. The latter case contradicts Lemma 5.1, since $L_i - b$ is an endpath of \mathcal{S} . However, then the former case implies that $X \cap V(L_i - b)$ assumes the unique structure shown in Figure 5.9 for all $i \in \{1, \dots, r\}$ and that a_j^* is odd for some $j \in \{1, \dots, r\}$. But then it is easy to see that there is at least one vertex of $L_j - b$ that is not defended by any vertex of X , again a contradiction. ■

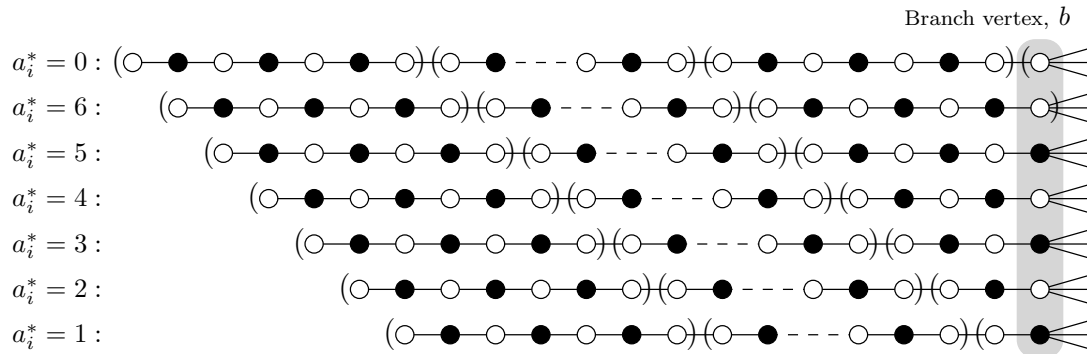
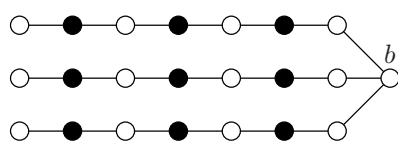


FIGURE 5.9: The solid vertices included in the set $\Lambda_i(\mathcal{S})$ for different values of a_i^* .

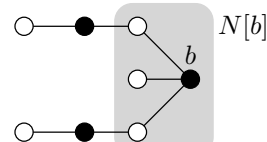
The following result shows that the set $\Lambda(\mathcal{S})$ can be augmented very slightly to form a minimum secure dominating set of a spider \mathcal{S} .

Lemma 5.2 *The set $\Lambda(\mathcal{S})$ in (5.7) securely dominates all the reduced legs of a spider \mathcal{S} with anchor b . Moreover, there exists a subset $N^* \subseteq N[b]$ such that $\bar{\Lambda}(\mathcal{S}) = \Lambda(\mathcal{S}) \cup N^*$ is a minimum secure dominating set of \mathcal{S} .*

Proof: Note that the anchor b is not necessarily defended by any vertex in $\Lambda(\mathcal{S})$ — see Figure 5.10(a) for a special case in point. Furthermore, when viewed in the context of the entire spider, the vertex of the leg $L_i - b$ that is adjacent to the anchor may not be defended, even if it is defended when viewing the extended leg L_i as a path in isolation (because the anchor may be required to defend a vertex in another leg of the spider — see the spider in Figure 5.10(b) for an example of this phenomenon). Every vertex of the reduced leg L'_i is, however, dominated securely by $\Lambda_i(\mathcal{S})$ even when viewed in the context of the entire spider, and it follows by Lemma 5.1 that $\Lambda_i(\mathcal{S})$ is a smallest set with this property.



(a) The anchor b is not defended



(b) Not all vertices in $N[b]$ are defended

FIGURE 5.10: Solid vertices are included in $\Lambda(\mathcal{S})$. (a) A spider $\mathcal{S}(7, 7, 7)$ with characteristic vector $\mathbf{a}^* = (0, 0, 0)$ for which the anchor is not defended by any vertex in $\Lambda(\mathcal{S})$. (b) The spider $\mathcal{S}(3, 3, 1)$ with characteristic vector $\mathbf{a}^* = (3, 3, 1)$ in which some vertices in $N[b]$ are not defended.

Suppose X is a minimum secure dominating set of the entire spider. Another secure dominating set X' of \mathcal{S} with cardinality at most $|X|$ may be formed by “shifting” the vertices of X towards the anchor along each reduced leg of the spider, until these vertices conform to the optimum pattern in Figure 5.8. If, during this shifting process, vertices are shifted “out of” a reduced leg into the closed neighbourhood $N[b]$ of b , then these vertices are merely superimposed on top of the pattern $\Lambda(\mathcal{S}) \cap N[b]$. If, however, vertices are shifted “out of” an extended leg of the spider

altogether, then these vertices are discarded (disregarded). The sets $\Lambda(\mathcal{S})$ and X' differ only within $N[b]$, and, because $|X'| \leq |X|$, it follows that X' is a minimum secure dominating set of the spider with the property that $\Lambda(\mathcal{S}) \cap N[b] \subseteq X' \cap N[b]$. Therefore the statement of the lemma holds with $N^* = X' - \Lambda(\mathcal{S})$. ■

The following result is a characterisation of when the set $\Lambda(\mathcal{S})$ in (5.7) securely dominates the entire spider \mathcal{S} , i.e. when $N^* = \emptyset$ in Lemma 5.2.

Lemma 5.3 (Characterisation of when $\Lambda(\mathcal{S})$ is a secure dominating set)

Let \mathcal{S} be a spider with characteristic vector $\mathbf{a}^* = (a_1^*, \dots, a_r^*)$. Then the set $\Lambda(\mathcal{S})$ in (5.7) is a minimum secure dominating set of \mathcal{S} if and only if:

- (a) $3 \notin \mathbf{a}^*$ and \mathbf{a}^* contains at most one unit entry and at least one odd entry, or
- (b) $3 \in \mathbf{a}^*$ and $1 \notin \mathbf{a}^*$, or
- (c) $1, 3, 5 \notin \mathbf{a}^*$ and $6 \in \mathbf{a}^*$, or
- (d) $1, 3, 5, 6 \notin \mathbf{a}^*$ and $4 \in \mathbf{a}^*$ and \mathbf{a}^* contains at most $r - 2$ zero entries.

Proof: By Lemma 5.2 it need only be verified that $\Lambda(\mathcal{S})$ securely dominates the closed neighbourhood $N[b]$ of the anchor b of \mathcal{S} in, and only in, the above four cases. Note that if $\Lambda(\mathcal{S})$ securely dominates \mathcal{S} , then $\Lambda(\mathcal{S})$ is necessarily a minimum secure dominating set of \mathcal{S} by Corollary 5.1. Denote the neighbour of b in the extended leg L_i of \mathcal{S} by v_i for all $1 \leq i \leq r$ throughout the proof of this lemma. Consider two mutually exclusive cases, namely where the characteristic vector \mathbf{a}^* contains some odd entries, and where it contains only even entries.

Case 1: \mathbf{a}^ contains some odd entries.* In this case $b \in \Lambda(\mathcal{S})$ and hence it must be verified that the open neighbourhood $N(b)$ is securely dominated by $\Lambda(\mathcal{S})$ in, and only in, the above cases. The following two subcases (corresponding to parts (a) and (b) of the lemma) are distinguished:

Subcase 1(a): $3 \notin \mathbf{a}^$.* If it also holds that $1 \notin \mathbf{a}^*$, then the only odd entries in \mathbf{a}^* have the value 5. If $a_i^* = 2, 4$ or 6 , then $v_i \in \Lambda(\mathcal{S})$. However, if $a_i^* = 0$ or 5 , then $v_i \notin \Lambda(\mathcal{S})$ but v_i is defended by its neighbour in $L_i - b$, and so $\Lambda(\mathcal{S})$ is a secure dominating set of \mathcal{S} .

If \mathbf{a}^* contains exactly one unit entry, $a_j^* = 1$ (say), then b defends v_j , all the other vertices in $N(b)$ are defended as above, and $\Lambda(\mathcal{S})$ is again a secure dominating set of \mathcal{S} .

If \mathbf{a}^* contains at least two unit entries, $a_j^* = 1$ and $a_k^* = 1$ (say), then b can defend neither v_j nor v_k , as neither swap sets are dominating sets of \mathcal{S} .

Subcase 1(b): $3 \in \mathbf{a}^$.* If $1 \notin \mathbf{a}^*$, then the only odd entries in \mathbf{a}^* are the values 3 and (possibly) 5. If $a_i^* = 3$, then b defends v_i . If $a_i^* = 2, 4$ or 6 , then v_i defends itself. Finally, if $a_i^* = 0$ or 5 , then v_i is defended by its neighbour in $L_i - b$. $\Lambda(\mathcal{S})$ is therefore a secure dominating set of \mathcal{S} .

If \mathbf{a}^* contains both a unit entry and the value 3, $a_j^* = 1$ and $a_k^* = 3$ (say), then b cannot defend both v_j and v_k , and so $\Lambda(\mathcal{S})$ is not a secure dominating set of \mathcal{S} .

Case 2: \mathbf{a}^ contains only even entries.* If $a_i^* = 2, 4$ or 6 , then $v_i \in \Lambda(\mathcal{S})$. Furthermore, if $a_i^* = 0$, then v_i is defended by its neighbour in $L_i - b$. Since $b \notin \Lambda(\mathcal{S})$, it must be verified that b is defended by some vertex in $\Lambda(\mathcal{S})$ in, and only in, the above cases. Again two subcases (corresponding to parts (c) and (d) of the lemma) are distinguished:

Subcase 2(a): $6 \in \mathbf{a}^$.* Suppose $a_j^* = 6$. Then b is defended by v_j , and so $\Lambda(\mathcal{S})$ is a secure dominating set of \mathcal{S} .

Subcase 2(b): $6 \notin \mathbf{a}^$.* If \mathbf{a}^* contains the value 4, $a_j^* = 4$ (say), and \mathbf{a}^* contains at most $r - 2$ zero entries, then b is defended by v_j , and so $\Lambda(\mathcal{S})$ is a secure dominating set of \mathcal{S} .

If \mathbf{a}^* contains the value 4, $a_j^* = 4$ (say), and $r - 1$ zero entries, then the neighbour of v_j in $L_j - b$ is not defended. Hence, $\Lambda(\mathcal{S})$ is not a secure dominating set of \mathcal{S} .

Finally, if $4 \notin \mathbf{a}^*$, then the only entries in \mathbf{a}^* are the values 0 and/or 2, and for each neighbour $v_i \in \Lambda(\mathcal{S})$ of b , the swap set $(\Lambda(\mathcal{S}) - \{v_i\}) \cup \{b\}$ does not dominate \mathcal{S} ; hence $\Lambda(\mathcal{S})$ is not a secure dominating set of \mathcal{S} . ■

Let

$$R_1(\mathcal{S}) = \bigcup_{i=1}^r \{v_i : a_i^* = 1\}. \quad (5.8)$$

If $R_1(\mathcal{S}) \neq \emptyset$, let x be an arbitrary vertex of $R_1(\mathcal{S})$ and let

$$R'_1(\mathcal{S}) = R_1(\mathcal{S}) - \{x\}. \quad (5.9)$$

The following result contains a specification of how the set $\Lambda(\mathcal{S})$ should be augmented to form a secure dominating set $\bar{\Lambda}(\mathcal{S})$ of the spider \mathcal{S} in the four subcases of the proof of Lemma 5.3 when $\Lambda(\mathcal{S})$ itself is *not* a secure dominating set of the spider.

Lemma 5.4 (If $\Lambda(\mathcal{S})$ is not a secure dominating set, how to extend it)

Let \mathcal{S} be a spider with characteristic vector $\mathbf{a}^* = (a_1^*, \dots, a_r^*)$ and anchor b . Furthermore, let $R_1(\mathcal{S})$ and $R'_1(\mathcal{S})$ be the sets in (5.8) and (5.9).

- (a) If $3 \notin \mathbf{a}^*$, but \mathbf{a}^* contains at least 2 unit entries, then $\bar{\Lambda}(\mathcal{S}) = \Lambda(\mathcal{S}) \cup R'_1(\mathcal{S})$ is a minimum secure dominating set of \mathcal{S} .
- (b) If $1, 3 \in \mathbf{a}^*$, then $\bar{\Lambda}(\mathcal{S}) = \Lambda(\mathcal{S}) \cup R_1(\mathcal{S})$ is a minimum secure dominating set of \mathcal{S} .
- (c) If $1, 3, 4, 5, 6 \notin \mathbf{a}^*$, then $\bar{\Lambda}(\mathcal{S}) = \Lambda(\mathcal{S}) \cup \{b\}$ is a minimum secure dominating set of \mathcal{S} .
- (d) If $1, 3, 5, 6 \notin \mathbf{a}^*$ and \mathbf{a}^* contains at least $r - 1$ zero entries, then $\bar{\Lambda}(\mathcal{S}) = \Lambda(\mathcal{S}) \cup \{b\}$ is a minimum secure dominating set of \mathcal{S} .

Proof: (a) If $3 \notin \mathbf{a}^*$ but \mathbf{a}^* contains at least two unit entries, then $b \in \Lambda(\mathcal{S})$ and every vertex of $N[b] - R_1(\mathcal{S})$ can form a dominating swap set of \mathcal{S} with some vertex in $\Lambda(\mathcal{S})$ other than b . Since no swap set $(\Lambda(\mathcal{S}) - \{b\}) \cup \{v\}$ dominates \mathcal{S} , where $v \in R_1(\mathcal{S})$, it follows that $\Lambda(\mathcal{S}) \cup R'_1(\mathcal{S})$ is a minimum secure dominating set of \mathcal{S} by Lemma 5.2.

(b) If $1, 3 \in \mathbf{a}^*$, then $b \in \Lambda(\mathcal{S})$ is required to defend all its neighbours in legs for which $a_i^* = 3$. Moreover, all the neighbours of b in legs for which a_i^* is even are defended by vertices in $\Lambda(\mathcal{S})$ other than b . Since every vertex $v \in R_1(\mathcal{S})$ can be defended by itself only, $\Lambda(\mathcal{S}) \cup R_1(\mathcal{S})$ is a minimum secure dominating set of \mathcal{S} by Lemma 5.2.

(c) If $1, 3, 4, 5, 6 \notin \mathbf{a}^*$, then \mathbf{a}^* only contains the values 0 and/or 2, and so b is the only vertex in $N[b]$ that is not defended by any vertex in $\Lambda(\mathcal{S})$. Since $\Lambda(\mathcal{S}) \cup \{b\}$ is, however, a secure dominating set of \mathcal{S} , it is also a minimum secure dominating set of \mathcal{S} by Lemma 5.2.

(d) If $1, 3, 5, 6 \notin \mathbf{a}^*$ and \mathbf{a}^* contains at least $r - 1$ zeros, then $a_j^* = 0, 2$ or 4 (say) and $a_i^* = 0$ for all $i \neq j$. In this case b is only dominated (but not defended) by $v_j \in \Lambda(\mathcal{S})$. Since $\Lambda(\mathcal{S}) \cup \{b\}$ is, however, a secure dominating set of \mathcal{S} , it is also a minimum secure dominating set by Lemma 5.2. ■

The manner in which $\Lambda(\mathcal{S})$ in (5.7) is augmented to form a minimum secure dominating set $\bar{\Lambda}(\mathcal{S})$ of a spider \mathcal{S} according to Lemma 5.4 is illustrated by two examples in Figure 5.11. The spider $\mathcal{S}(3, 1, 8, 2)$ in Figure 5.11(a) has characteristic vector $\mathbf{a}^* = (3, 1, 1, 2)$. The set $\Lambda(\mathcal{S})$ for this

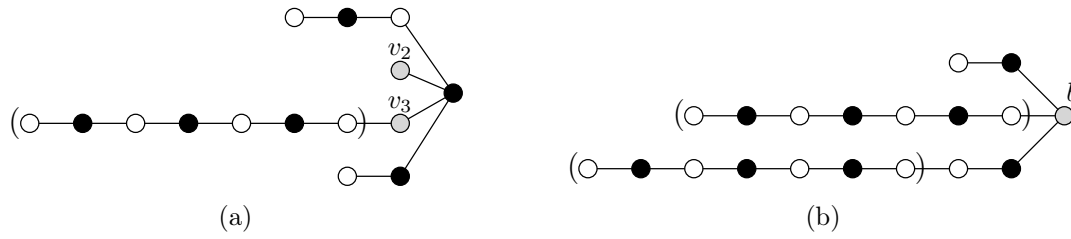


FIGURE 5.11: Solid vertices are included in the set $\Lambda(\mathcal{S})$ in (5.7). The union of the sets of solid and grey vertices represent the minimum secure dominating set $\bar{\Lambda}(\mathcal{S})$ in Lemma 5.4 for the spider in each case. (a) The spider $\mathcal{S}(3, 1, 8, 2)$ with characteristic vector $\mathbf{a}^* = (3, 1, 1, 2)$ for which the vertices $\{v_2, v_3\}$ in $N(b)$ are not securely dominated by $\Lambda(\mathcal{S})$. (b) The spider $\mathcal{S}(2, 7, 9)$ with characteristic vector $\mathbf{a}^* = (2, 0, 2)$ for which the anchor is not securely dominated by $\Lambda(\mathcal{S})$.

spider is denoted by the solid vertices in the figure. The set $\{v_2, v_3\}$ is added to $\Lambda(\mathcal{S})$ to yield the minimum secure dominating set $\bar{\Lambda}(\mathcal{S})$ according to Lemma 5.4(b), which is indicated by the solid and grey vertices combined. The spider $\mathcal{S}(2, 7, 9)$ in Figure 5.11(b) has characteristic vector $\mathbf{a}^* = (2, 0, 2)$. Again the set $\Lambda(\mathcal{S})$ is denoted by the solid vertices. The anchor b is added to $\Lambda(\mathcal{S})$ to deliver the minimum secure dominating set $\bar{\Lambda}(\mathcal{S})$ by Lemma 5.4(c), which is again indicated in the figure by the solid and grey vertices combined.

5.2.2 From spiders to trees in general

Let T be a tree that is not a spider. A *pendent spider* \mathcal{S} of T is a subgraph of T that is a spider with anchor b whose endpaths are also endpaths of T and for which $\deg_{\mathcal{S}}(b) = \deg_T(b) - 1$. All the pendent spiders of the tree T_1 in Figure 5.12(a) are highlighted in grey in Figure 5.12(b). Let v be the (only) neighbour of b in T that is not also in \mathcal{S} . The *pendent path* in T associated with \mathcal{S} is the path starting at v and continuing into $T - \mathcal{S}$ up to (and including) the first branch vertex encountered in $T - \mathcal{S}$. The pendent paths of the pendent spiders in T_1 are shown in boldface in Figure 5.12(b).

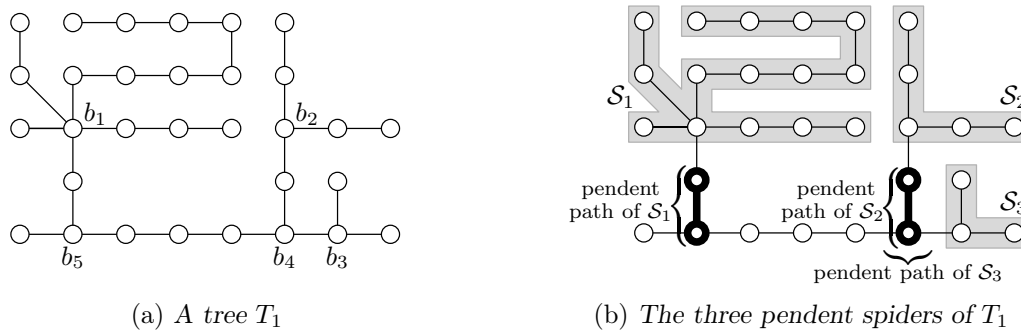


FIGURE 5.12: (a) A tree T_1 containing five branch vertices b_1, b_2, b_3, b_4 and b_5 . (b) The three pendent spiders $\mathcal{S}_1, \mathcal{S}_2$ and \mathcal{S}_3 of T_1 are highlighted in grey, while the corresponding pendent paths are shown in boldface.

An approach very similar to the one described in §5.2.1 is adopted for constructing a minimum secure dominating set for a pendent spider, as described in Lemma 5.4. The only difference between a minimum secure dominating set of a spider and that of a pendent spider involves possibly defending the anchor of the latter from its pendent path rather than from within the pendent spider.

Let

$$\bar{\Lambda}^*(\mathcal{S}) = \begin{cases} \Lambda(\mathcal{S}) \text{ as defined in (5.7) if any one of the four conditions in Lemma 5.3} \\ \text{holds and } \mathcal{S} \text{ is a pendent spider of } T, \\ \bar{\Lambda}(\mathcal{S}) \text{ as defined in Lemma 5.4(a) or 5.4(b) if any one of the} \\ \text{corresponding conditions holds and } \mathcal{S} \text{ is a pendent spider of } T, \\ \bar{\Lambda}(\mathcal{S}) \text{ as defined in Lemma 5.4(a), 5.4(b), 5.4(c) or 5.4(d) if any one of} \\ \text{the corresponding conditions holds and } T = \mathcal{S} \text{ is a spider.} \end{cases} \quad (5.10)$$

The following result follows as a corollary of Lemma 5.1.

Corollary 5.2 *If $\mathcal{S} = T$ is a spider or \mathcal{S} is a pendent spider of a tree T which is itself not a spider, then no secure dominating set of T contains fewer than $|\bar{\Lambda}^*(\mathcal{S})|$ vertices from \mathcal{S} .*

The proof of Corollary 5.2 is virtually identical to that of Corollary 5.1 and is hence omitted.

The approach towards constructing a minimum secure dominating set for an arbitrary tree T proposed in the following subsection hinges on repeatedly pruning away pendent spiders from T after having dominated these pendent spiders securely, until only a final spider remains. The next result shows that this approach is viable.

Lemma 5.5 *Let \mathcal{S} be a pendent spider of a tree T . Then there exists a minimum secure dominating set of T containing $\bar{\Lambda}^*(\mathcal{S})$ as subset.*

Proof: Let X be a smallest possible subset of vertices from $T - \mathcal{S}$ for which $\bar{\Lambda}^*(\mathcal{S}) \cup X$ is a secure dominating set of T . If $\bar{\Lambda}^*(\mathcal{S}) \cup X$ is, in fact, a minimum secure dominating set of T , then the proof is complete. Otherwise, there is a secure dominating set Y of T such that $|Y| \leq |\bar{\Lambda}^*(\mathcal{S}) \cup X|$. But since $|Y \cap V(\mathcal{S})| \geq |\bar{\Lambda}^*(\mathcal{S})|$ by Corollary 5.2, it follows that $|Y \cap V(T - \mathcal{S})| \leq |X|$, contradicting the minimality of X . ■

The continuation of the secure dominating set pattern in Figure 5.8 from the anchor b of a pendent spider \mathcal{S} with characteristic vector $\mathbf{a}^* = (a_1^*, \dots, a_{r-1}^*)$ of a tree T into the rest of T along the pendent path of \mathcal{S} is based on the assignment of a list of labels to each vertex of \mathcal{S} . Initially these lists of labels are all empty. The label list of each vertex u in the extended leg L_i of \mathcal{S} is appended by including the distance modulo 7 from the leaf in L_i to u in the list, for all $i = 1, \dots, r - 1$. Note, therefore, that the label list of b contains the values a_1^*, \dots, a_{r-1}^* , while the label lists of every other vertex of \mathcal{S} contains a single entry, as illustrated in Figure 5.13.

The vertices of the endpaths of \mathcal{S} are included in the construction of the secure dominating set of T if they form part of the set $\bar{\Lambda}^*(\mathcal{S})$ (*i.e.* their label lists contain odd entries). The inclusion or otherwise of the anchor b in the secure dominating set of T is decided by assigning an *effective label* to b , denoted by $\ell(b)$, which depends on the characteristic vector of \mathcal{S} and hence the label list of b . The value of this effective label is determined by traversing the decision tree in Figure 5.14. This decision tree has been designed to determine the necessity of including the neighbour of the anchor of a pendent spider on its pendent path in a secure dominating set X of T by exploring the valid swap sets of X involving v .

The next result expresses the value of the effective label $\ell(b)$ of the anchor b of a pendent spider directly in terms of the spider's characteristic vector, thereby alleviating the need to traverse the decision tree in Figure 5.14 repeatedly in an algorithm for secure domination of trees.

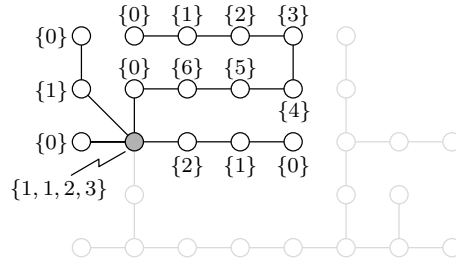


FIGURE 5.13: The label lists of the vertices of the pendent spider \mathcal{S}_1 of the tree T_1 in Figure 5.12(b) with characteristic vector $\mathbf{a}^* = (1, 3, 2, 1)$. The anchor is indicated in grey and is the only vertex with more than one entry in its label list.

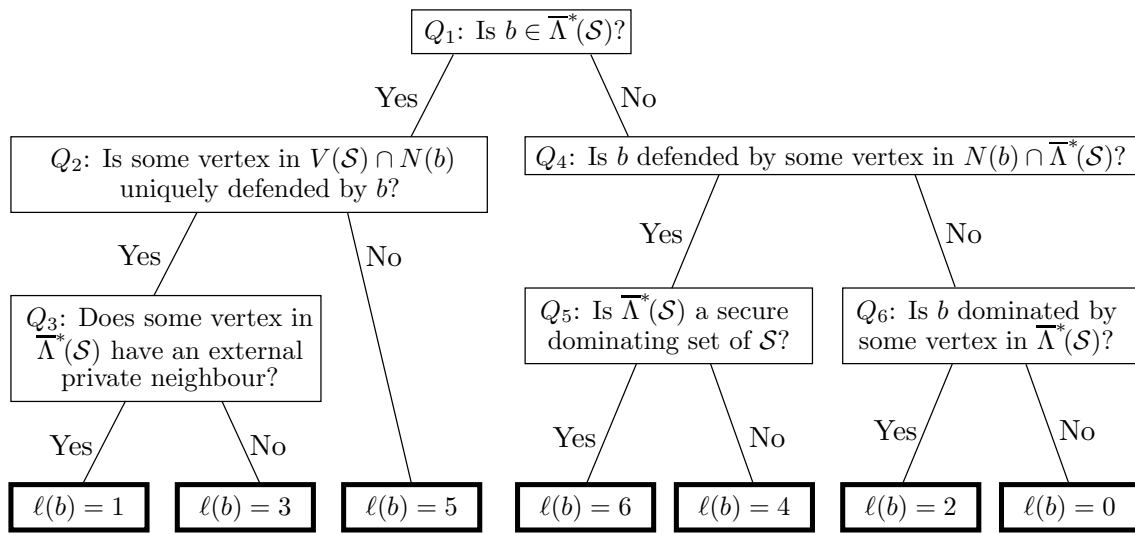


FIGURE 5.14: Decision tree of binary questions for determining the effective label $\ell(b)$ of the anchor b of a pendent spider \mathcal{S} .

Lemma 5.6 (The effective label of the anchor of a pendent spider)

Let \mathcal{S} be a pendent spider of a tree with characteristic vector $\mathbf{a}^* = (a_1^*, \dots, a_{r-1}^*)$ and an anchor b with effective label $\ell(b)$, as determined by the decision tree in Figure 5.14. If $1, 3, 5, 6 \notin \mathbf{a}^*$, but $4 \in \mathbf{a}^*$ and \mathbf{a}^* contains at most $r - 3$ zero entries, then $\ell(b) = 6$. Otherwise $\ell(b)$ is the first element of the ordered set $\{3, 1, 5, 6, 4, 2, 0\}$ also in \mathbf{a}^* .

Proof: Denote the neighbour of b in the extended leg L_i of \mathcal{S} by v_i for all $i \in \{1, \dots, r - 1\}$ throughout the proof of this lemma. It is systematically verified that the value of $\ell(b)$ in the statement of the lemma corresponds to the value of the effective label as determined by the decision tree in Figure 5.14. Consider two mutually exclusive cases, namely where the characteristic vector \mathbf{a}^* contains only even entries, and where it contains some odd entries.

Case 1: \mathbf{a}^ contains only even entries.* In this case $b \notin \overline{L}^*(\mathcal{S})$ and $\ell(b) \neq 1, 3, 5$, as can be seen in the decision tree.

If $\mathbf{a}^* = \mathbf{0}$, then b is neither defended nor dominated by any vertex in \mathcal{S} . From the decision tree it therefore follows that $\ell(b) = 0$.

If $4, 6 \notin \mathbf{a}^*$ and $2 \in \mathbf{a}^*$, then b is dominated, but not defended, in which case it follows from the decision tree that $\ell(b) = 2$.

If $6 \notin \mathbf{a}^*$ and $4 \in \mathbf{a}^*$, consider the following three subcases.

Subcase i: The vector \mathbf{a}^* contains at least two 4-entries, $a_i^* = 4$ and $a_j^* = 4$ (say). Since b is defended by v_i and v_j , it remains dominated by v_i or v_j for any swap set of $\overline{\Lambda}^*(\mathcal{S})$, implying that $\overline{\Lambda}^*(\mathcal{S})$ is a secure dominating set of \mathcal{S} and so $\ell(b) = 6$ according to the decision tree.

Subcase ii: The vector \mathbf{a}^* contains both a single 4-entry and a 2-entry, $a_i^* = 4$ and $a_j^* = 2$ (say). In this case again $\ell(b) = 6$ according to the decision tree, since b is defended by v_i and dominated by v_j , ensuring that $\overline{\Lambda}^*(\mathcal{S})$ is a secure dominating set of \mathcal{S} .

Subcase iii: The vector \mathbf{a}^* contains a single 4-entry, $a_i^* = 4$ (say), and no 2-entry. In this case $\ell(b) = 4$ according to the decision tree, since b is defended by v_i , but $\overline{\Lambda}^*(\mathcal{S})$ is not a secure dominating set of \mathcal{S} .

If $6 \in \mathbf{a}^*$, then $\ell(b) = 6$ according to the decision tree, since b is defended by some vertex in $N(b) \cap \overline{\Lambda}^*(\mathcal{S})$ and $\overline{\Lambda}^*(\mathcal{S})$ is a secure dominating set of \mathcal{S} .

Case 2: \mathbf{a}^ contains some odd entries.* In this case $b \in \overline{\Lambda}^*(\mathcal{S})$ and $\ell(b) \neq 0, 2, 4, 6$, as can be seen in the decision tree.

If $1, 3 \notin \mathbf{a}^*$, then $\ell(b) = 5$ according to the decision tree, as no vertex in $N(b)$ is uniquely defended.

If $3 \notin \mathbf{a}^*$ and $1 \in \mathbf{a}^*$, then b has a private neighbour external to $\overline{\Lambda}^*(\mathcal{S})$ which is uniquely defended by b and hence $\ell(b) = 1$ according to the decision tree.

If $3 \in \mathbf{a}^*$, then there is a vertex in $V(\mathcal{S}) \cap N(b)$ that is uniquely defended by b and no vertex in $\overline{\Lambda}^*(\mathcal{S})$ has an external private neighbour. Hence $\ell(b) = 3$ according to the decision tree. ■

The function of the effective label $\ell(b)$ is that it serves as a starting point for the secure dominating set pattern in Figure 5.8 along the pendent path of \mathcal{S} into the rest of T , as illustrated in Figure 5.15. The reason for choosing the effective label according to the decision tree in Figure 5.14 is to ensure that the final secure dominating set constructed for T is as small as possible. To achieve this, the occurrence of two adjacent vertices along the pendent path of \mathcal{S} that are not in the secure dominating set of T should be induced as soon as possible when traversing the pendent path from the direction of b .

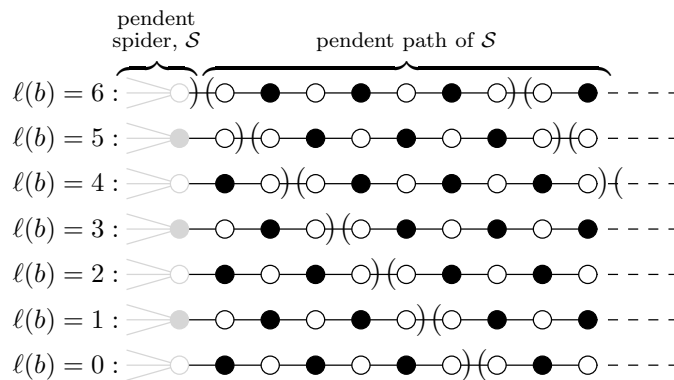


FIGURE 5.15: Using the effective label, $\ell(b)$, to determine the starting point for the continuation of the secure dominating set pattern in Figure 5.8 along the pendent path of a pendent spider \mathcal{S} of a tree into the rest of the tree.

The following result shows that the continuation of the secure dominating set pattern in Figure 5.8 from b into the rest of T along the pendent path of \mathcal{S} , using the value $\ell(b)$ as starting point for the pattern, is a viable approach towards determining a minimum secure dominating set of T .

Lemma 5.7 *Let \mathcal{S} be a pendent spider of a tree T , let b be the anchor of \mathcal{S} and let $\ell(b)$ be the effective label of b , as determined by traversing the decision tree in Figure 5.14. Then no starting point for the continuation of the secure dominating set pattern in Figure 5.8 along the pendent path of \mathcal{S} can result in a smaller secure dominating set of T than the starting point $\ell(b)$, illustrated in Figure 5.15.*

Proof: It follows from Lemma 5.5 that there exists a minimum secure dominating set X of T containing $\bar{\Lambda}^*(\mathcal{S})$ as subset and hence in which the label of the anchor b of \mathcal{S} has the value $\ell(b)$, as determined in the decision tree of Figure 5.14. It is easy to verify (by using this decision tree) that the anchor of a single-leg pendent spider \mathcal{S}' with (extended) leg length ℓ and anchor b also has effective label $\ell(b)$. The pendent spider \mathcal{S} in T may therefore be replaced by \mathcal{S}' to obtain a new tree T' for which $\Lambda(\mathcal{S}') \cup (X - \bar{\Lambda}^*(\mathcal{S}))$ is a minimum secure dominating set. Hence it follows that the anchor b of \mathcal{S} can deliver no other starting point for the secure dominating set continuation pattern in Figure 5.8 along the pendent path of \mathcal{S} into the rest of T which results in a smaller secure dominating set of T than the one obtained by using $\ell(b)$ as starting point for this pattern. ■

As mentioned above, the algorithmic approach proposed in the following subsection towards computing the value of $\gamma_s(T)$ for a tree T is to select a pendent spider \mathcal{S} of T , compute a secure dominating set for this spider \mathcal{S} , prune away the spider, and repeat the process for the smaller tree thus formed, until a tree is reached that is itself a spider for which the set $\bar{\Lambda}(\mathcal{S})$, as defined in Lemma 5.4, is a minimum secure dominating set. The value of $\gamma_s(T)$ is then the sum of the cardinalities of the minimum secure dominating sets of the spiders pruned away and that of the final spider, as established in the following result.

Theorem 5.2 *Let $T_1, T_2, \dots, T_\Omega$ be a sequence of trees in which T_{i+1} is formed from T_i by pruning away a pendent spider \mathcal{S}_i from T_i for all $i = 1, \dots, \Omega - 1$ until a spider $\mathcal{S}_\Omega = T_\Omega$ is reached. Then*

$$\gamma_s(T_1) = \sum_{i=1}^{\Omega} |\bar{\Lambda}^*(\mathcal{S}_i)|.$$

Proof: It is first shown that

$$\gamma_s(T_i) = \gamma_s(T'_i) + |\bar{\Lambda}^*(\mathcal{S}_i)| - \lceil 3\ell(b_i)/7 \rceil, \quad i = 1, \dots, \Omega - 1, \quad (5.11)$$

where T'_i is formed by replacing \mathcal{S}_i by a path $\mathcal{P}_{1+\ell(b_i)}$ of length $\ell(b_i)$ in T_i and where b_i is the anchor of \mathcal{S}_i , as illustrated in Figure 5.16 and described in the proof of Lemma 5.7. It follows by Proposition 3.9 and Lemma 5.7 that there exist a minimum secure dominating set of T'_i containing $\lceil 3\ell(b_i)/7 \rceil$ vertices of $\mathcal{P}_{1+\ell(b_i)}$ and a minimum secure dominating set of T_i containing $|\bar{\Lambda}^*(\mathcal{S}_i)|$ vertices of \mathcal{S}_i . Hence there exists a set of vertices $Y_i \subseteq V(T_i - \mathcal{S}_i)$ such that $\gamma_s(T_i) = |\bar{\Lambda}^*(\mathcal{S}_i)| + |Y_i|$. Since $T_i - \mathcal{S}_i = T'_i - \mathcal{P}_{1+\ell(b_i)}$, it holds that $\gamma_s(T'_i) = \lceil 3\ell(b_i)/7 \rceil + |Y_i|$, and so (5.11) follows.

It follows from (5.11) that

$$\gamma_s(T_i) = \gamma_s(T_{i+1}) + \left\lceil \frac{3\ell(b_i)}{7} \right\rceil + |\bar{\Lambda}^*(\mathcal{S}_i)| - \left\lceil \frac{3\ell(b_i)}{7} \right\rceil = \gamma_s(T_{i+1}) + |\bar{\Lambda}^*(\mathcal{S}_i)|,$$

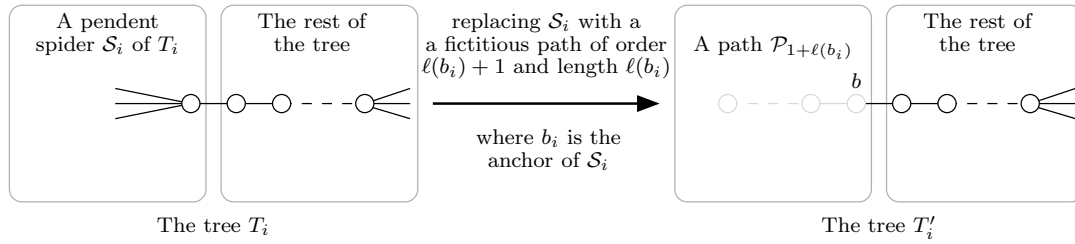


FIGURE 5.16: Forming a tree T'_i by replacing a pendent spider \mathcal{S}_i of T_i by a path $\mathcal{P}_{1+\ell(b_i)}$ of length $\ell(b_i)$ where b_i is the anchor of \mathcal{S}_i .

since $T_{i+1} = T'_i - \mathcal{P}_{1+\ell(b_i)}$, from which it follows that

$$|\bar{\Lambda}^*(\mathcal{S}_i)| = \gamma_s(T_i) - \gamma_s(T_{i+1}), \quad i = 1, \dots, \Omega - 1. \quad (5.12)$$

Therefore,

$$\begin{aligned} \sum_{i=1}^{\Omega} |\bar{\Lambda}^*(\mathcal{S}_i)| &= |\bar{\Lambda}^*(\mathcal{S}_{\Omega})| + \sum_{i=1}^{\Omega-1} |\bar{\Lambda}^*(\mathcal{S}_i)| \\ &= |\bar{\Lambda}^*(\mathcal{S}_{\Omega})| + \sum_{i=1}^{\Omega-1} [\gamma_s(T_i) - \gamma_s(T_{i+1})] \quad \text{by (5.12)} \\ &= |\bar{\Lambda}^*(\mathcal{S}_{\Omega})| + \gamma_s(T_1) - \gamma_s(T_{\Omega}) \\ &= |\bar{\Lambda}^*(\mathcal{S}_{\Omega})| + \gamma_s(T_1) - |\bar{\Lambda}^*(\mathcal{S}_{\Omega})| \\ &= \gamma_s(T_1), \end{aligned}$$

as desired. ■

Consider, as an example, the tree T_1 in Figure 5.17(a) with pendent spider \mathcal{S}_1 , which has the characteristic vector $(1, 3, 2, 1)$. The set $\bar{\Lambda}^*(\mathcal{S}_1)$ is indicated by the solid vertices in the figure and has cardinality $|\bar{\Lambda}^*(\mathcal{S}_1)| = 8$. The label list of the anchor b_1 is $\{1, 3, 2, 1\}$. The effective label $\ell(b)$ is therefore 3 according to Lemma 5.6. This value represents a starting point for the secure dominating set pattern in Figure 5.8 into $T_2 = T_1 - \mathcal{S}_1$ along the pendent path of \mathcal{S}_1 , as shown in Figure 5.17(b). This pendent path is, of course, a leg of another pendent spider in T_2 , which will be securely dominated later. The pendent spider \mathcal{S}_2 in the tree T_2 has the characteristic vector $(2, 2)$ and the set $\bar{\Lambda}^*(\mathcal{S}_2)$ has cardinality $|\bar{\Lambda}^*(\mathcal{S}_2)| = 2$, as indicated by the solid vertices in Figure 5.17(c). The label list of b_2 is $\{2, 2\}$ and so the effective label of b_2 is 2 by Lemma 5.6. The pendent spider \mathcal{S}_3 in $T_3 = T_2 - \mathcal{S}_2$ shown in Figure 5.17(c) has the characteristic vector $(1, 1)$, which delivers a label list $\{1, 1\}$ for the anchor b_3 and so the effective label of b_3 is 1. The set $\bar{\Lambda}^*(\mathcal{S}_3)$ is indicated by the $|\bar{\Lambda}^*(\mathcal{S}_3)| = 2$ solid vertices in the figure. The set $\bar{\Lambda}^*(\mathcal{S}_4)$ is indicated by the solid vertex in Figure 5.17(d). The tree $T_5 = \mathcal{S}_5 = T_4 - \mathcal{S}_4$ is a spider with characteristic vector $(1, 3, 5)$ for which the set $\bar{\Lambda}^*(\mathcal{S}_5)$ has cardinality $|\bar{\Lambda}^*(\mathcal{S}_5)| = 3$, as indicated by the solid vertices in Figure 5.17(e). A minimum secure dominating set of T_1 therefore has cardinality $\gamma_s(T_1) = |\bar{\Lambda}^*(\mathcal{S}_1)| + |\bar{\Lambda}^*(\mathcal{S}_2)| + |\bar{\Lambda}^*(\mathcal{S}_3)| + |\bar{\Lambda}^*(\mathcal{S}_4)| + |\bar{\Lambda}^*(\mathcal{S}_5)| = 8 + 2 + 2 + 1 + 3 = 16$ by Theorem 5.2, as indicated by the solid vertices in Figure 5.17(f).

5.2.3 Linear algorithmic implementation

An *ordering* of the vertices of a tree T of order n is an assignment of the indices $1, \dots, n$ to the vertices of T , one index to a vertex. A *canonical ordering* of the vertices of a rooted tree T is an

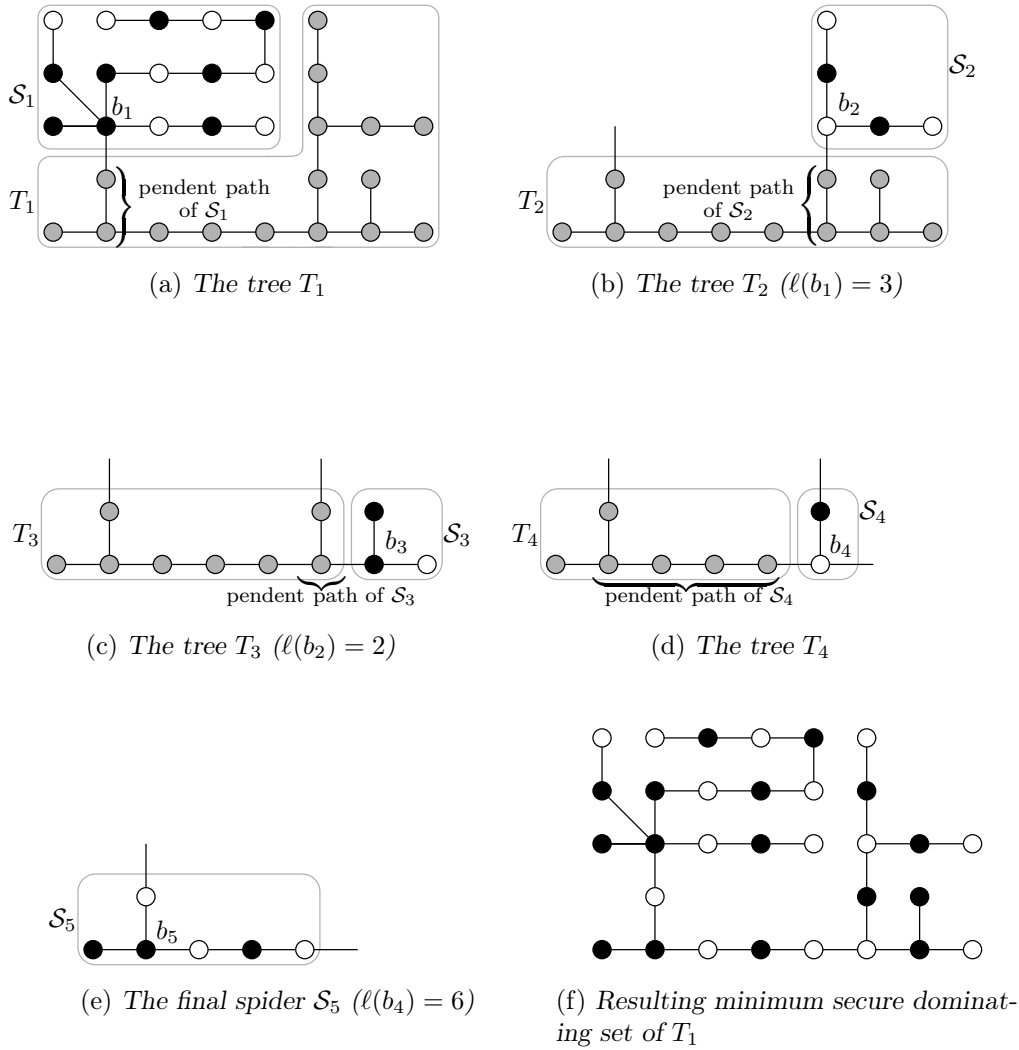


FIGURE 5.17: (a) The pendent spider \mathcal{S}_1 with anchor b_1 of the tree T_1 in Figure 5.12 connected by a pendent path to the rest of the tree. (b) The pendent spider \mathcal{S}_2 with anchor b_2 of the tree T_2 after having pruned away the pendent spider \mathcal{S}_1 . (c) The pendent spider \mathcal{S}_3 with anchor b_3 of the tree T_3 after having pruned away the pendent spider \mathcal{S}_2 . (d) The pendent spider \mathcal{S}_4 with anchor b_4 of the tree T_4 . (e) The final spider \mathcal{S}_5 with anchor b_5 . (f) A minimum secure dominating set of T_1 .

ordering of the vertices of T such that the index of the parent of vertex i , denoted by $\text{Parent}[i]$, is smaller than the vertex indexed i . The root of T therefore has index 1 and the special convention is adopted that the “index” of $\text{Parent}[1]$ is 0. Assume, without loss of generality, that the root of the tree T for which a minimum secure dominating set is sought is a branch vertex of T . Let $\text{Parent}[i_1, \dots, i_j]$ denote the set of vertex indices of the parents of the vertices indexed i_1, \dots, i_j . Then a canonical ordering of the vertices of T has the property that the tree induced by the set $\text{Parent}[1, \dots, k]$ is a subtree of the tree induced by the set $\text{Parent}[1, \dots, m]$ if $k \leq m$ [31, 78].

The algorithm for secure domination of a (rooted) tree T proposed here follows the approach described in §5.2.2 by traversing each vertex of T once while constructing a minimum secure dominating set for T , deciding whether or not that vertex should be included in the minimum secure dominating set and occasionally including a previously visited vertex in the minimum secure dominating set if certain conditions apply. In this traversal process the anchor of any pendent spider \mathcal{S} of T is a branch vertex of T (although not necessarily of \mathcal{S}). Six linear arrays

Algorithm 5.6: DefendTree

Input : A tree T represented by an array $\text{Parent}[1, \dots, n]$.
Output: A minimum secure dominating set of T , represented by a boolean array X .

```

1 for  $i \leftarrow 1$  to  $n$  do
2    $\text{A3Label}[i] \leftarrow \text{FALSE}$ ;
3    $X[i] \leftarrow \text{FALSE}$ ;
4    $\text{Labels}[i] \leftarrow [0, 0, 0, 0, 0, 0, 0]$ ;
5   if vertex  $i$  is a anchor of  $T$  then  $\text{Branch}[i] \leftarrow \text{TRUE}$ ;
6    $\text{Previous1Label}[i] \leftarrow 0$ ;
7 for  $i \leftarrow n$  down to 2 do
8    $\ell \leftarrow \text{EffectiveLabel}(\text{Labels}[i])$ ;
9   if  $\ell$  is odd then  $X[i] \leftarrow \text{TRUE}$ ;
10   $\text{Labels}[\text{Parent}[i], \ell(i) + 1 \pmod{7}] ++$ ;
11  if  $\ell = 2$  and  $\text{Branch}[\text{Parent}[i]]$  then
12     $\text{A3Label}[\text{Parent}[i]] \leftarrow \text{TRUE}$ ;
13     $\text{prev} \leftarrow \text{Previous1Label}[\text{Parent}[i]]$ ;
14    if  $\text{prev} > 0$  then  $X[\text{prev}] \leftarrow \text{TRUE}$ ;
15  if  $\ell = 0$  and  $\text{Branch}[\text{Parent}[i]]$  then
16    if  $\text{A3Label}[\text{Parent}[i]]$  then
17       $X[i] \leftarrow \text{TRUE}$ ;
18    else
19       $\text{prev} \leftarrow \text{Previous1Label}[\text{Parent}[i]]$ ;
20      if  $\text{prev} > 0$  then  $X[\text{prev}] \leftarrow \text{TRUE}$ ;
21       $\text{Previous1Label}[\text{Parent}[i]] \leftarrow i$ ;
22  $\ell \leftarrow \text{EffectiveLabel}(\text{Labels}[1])$ ;
23 if  $\ell$  is odd then  $X[1] \leftarrow \text{TRUE}$ ;
24 if  $\text{Labels}[1, j] = 0$  for  $j = 1, 3, 4, 5, 6$  then  $X[1] \leftarrow \text{TRUE}$ ;
25 if  $\text{Labels}[1, j] = 0$  for  $j = 1, 3, 5, 6$  and  $\text{Labels}[1, 0] \geq |\text{Labels}[1] - 1|$  then  $X[1] \leftarrow \text{TRUE}$ ;
26 if  $\ell = 3$  then
27    $\text{prev} \leftarrow \text{Previous1Label}[1]$ ;
28   if  $\text{prev} > 0$  then  $X[\text{prev}] \leftarrow \text{TRUE}$ ;
29 return  $X$ ;
```

are maintained during this traversal process:

$\text{Parent}[i]$ contains the index of the parent of vertex i in a canonical ordering of the vertices of T .

$\text{Branch}[i]$ contains the Boolean value **TRUE** if vertex i is a branch vertex of T , or the Boolean value **FALSE** otherwise.

$\text{Labels}[i]$ is initialised as an array of seven zeros. The entry in position $j \in \{0, \dots, 6\}$ of $\text{Labels}[i]$ is denoted by $\text{Labels}[i, j]$ and eventually represents the number of times the value j occurs in the label list of the vertex indexed i . This array is updated by increasing $\text{Labels}[\text{Parent}[i], \ell(i) + 1 \pmod{7}]$ by one as vertex i is visited during the traversal process. Note that if i is a branch vertex of T , then the weight (sum of the entries) of the array $\text{Labels}[i]$ will eventually be more than one, whereas the weight of $\text{Labels}[i]$ will eventually be one if i is not a branch vertex.

$\text{Previous1Label}[i]$ contains the last vertex index of the child of branch vertex i that caused the branch vertex to be assigned a value of 1 in its label list $\text{Labels}[i]$. It is necessary to keep track of this index because if a branch vertex i has a zero value in $\text{Labels}[i, 3]$, all its children with zero labels have to be included in the secure dominating set according to Lemma 5.6. It is therefore only possible to include i in the secure dominating set if the *next* 1-label is included in the list $\text{Labels}[i]$. However, if $\text{Labels}[i, 3]$ is already positive, then i can immediately be included in the secure dominating set by Lemma 5.6. This explains the necessity of the next linear array, namely A3Label .

$\text{A3Label}[i]$ is initialised to contain the Boolean value FALSE and is then updated to contain the Boolean value TRUE as soon as it becomes known that the value $\text{Labels}[i, 3]$ is positive and $\text{Branch}[i]$ is TRUE.

$X[i]$ is initialised to contain the boolean value FALSE and is updated to contain the boolean value TRUE if the vertex indexed i is at some point included in the minimum secure dominating set of T .

Steps 1–6 of Algorithm 5.6 initialise the arrays as described above. All the vertices of T are traversed in the for-loop spanning Steps 7–21, one vertex at a time, except for the root. The function **EffectiveLabel**($\text{Labels}[i]$) returns a single value in Step 8. If i is not a branch vertex of T , $\text{Labels}[i]$ will contain a single counter with a value of 1, in which case it returns the index of the counter. If i is a branch vertex, Lemma 5.6 is used to determine the effective label, depending on the entries of $\text{Labels}[i]$. In Step 9 the vertex i is added to the minimum secure dominating set if its effective label is odd. Step 10 assigns a new label to the parent of vertex i . Steps 11–14 ensure that all vertices which caused the assignment of a 1-label to a branch vertex i are included in the minimum secure dominating set, if $\text{A3Label}[i]$ is TRUE. Similarly, all but one vertex that causes the assignment of a 1-label to a branch vertex i is included in the minimum secure dominating set, if $\text{A3Label}[i]$ is FALSE, which occurs in Steps 15–21. Finally Steps 22–28 determine whether or not the root forms part of the minimum secure dominating set and ensure that all vertices which caused the assignment of a 1-label to the root are included in the minimum secure dominating set.

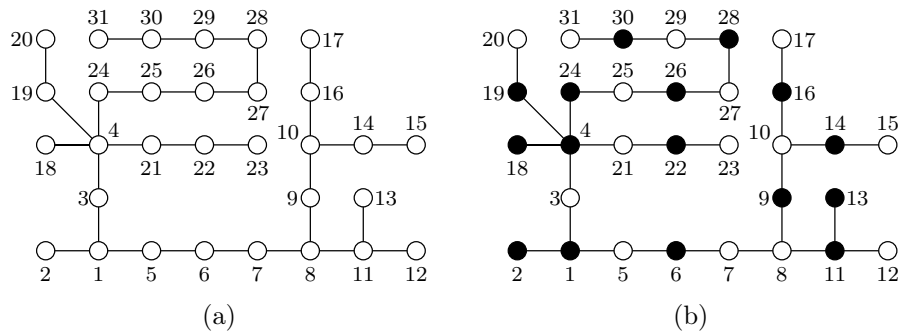


FIGURE 5.18: (a) A canonical ordering of the vertices of the tree T_1 in Figure 5.12(a). (b) The minimum secure dominating set of T_1 returned by Algorithm 5.6 is indicated by the solid vertices, as detailed in Table 5.4.

The minimum secure dominating set determined by Algorithm 5.6 for the tree T_1 of Figure 5.10 with canonical ordering as shown in Figure 5.18(a) is given in Figure 5.18(b). The six linear arrays maintained during execution of Algorithm 5.6 when applied to this tree, as well as the values returned by the function **EffectiveLabel**, are shown in Table 5.4.

This section is concluded with a result on the space and time complexities of Algorithm 5.6.

i	1	2	3	4	5	6	7	8
Parent[i]	0	1	1	3	1	5	6	7
Branch[i]	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE
Labels[i]	[0101010]	[1000000]	[0000100]	[0211000]	[0010000]	[0100000]	[1000000]	[0010100]
Previous1Label[i]	0	0	0	24	0	0	0	0
A3Label[i]	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
EffectiveLabel[i]	3	0	4	3	2	1	0	6
$X[i]$	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE
i	9	10	11	12	13	14	15	16
Parent[i]	8	9	8	11	11	10	14	10
Branch[i]	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
Labels[i]	[0001000]	[0020000]	[0200000]	[1000000]	[1000000]	[0100000]	[1000000]	[0100000]
Previous1Label[i]	0	0	12	0	0	0	0	0
A3Label[i]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
EffectiveLabel[i]	3	2	1	0	0	1	0	1
$X[i]$	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
i	17	18	19	20	21	22	23	24
Parent[i]	16	4	4	19	4	21	22	4
Branch[i]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Labels[i]	[1000000]	[1000000]	[0100000]	[1000000]	[0010000]	[0100000]	[1000000]	[1000000]
Previous1Label[i]	0	0	0	0	0	0	0	0
A3Label[i]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
EffectiveLabel[i]	0	0	1	0	2	1	0	0
$X[i]$	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
i	25	26	27	28	29	30	31	
Parent[i]	24	25	26	27	28	29	30	
Branch[i]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	
Labels[i]	[0000001]	[0000010]	[0000100]	[0001000]	[0010000]	[0100000]	[1000000]	
Previous1Label[i]	0	0	0	0	0	0	0	
A3Label[i]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	
EffectiveLabel[i]	6	5	4	3	2	1	0	
$X[i]$	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	

TABLE 5.4: The six linear arrays maintained during execution of Algorithm 5.6, as well as the output of the function **EffectiveLabel**, when applied to the tree T_1 in Figure 5.18.

Theorem 5.3 (Complexity of Algorithm 5.6) *If the input tree to Algorithm 5.6 has order n , then both the space complexity and the worst-case time complexity of Algorithm 5.6 are $\mathcal{O}(n)$.*

Proof: The function **EffectiveLabel**[i] returns the value in the label list if i is not a branch vertex of T or the value described in Lemma 5.6 if i is a branch vertex of T . The use of an array of length 7 to store the values in the list **Labels**[i] (and a similar approach towards storing the characteristic vector of each pendent spider) ensures that the function **EffectiveLabel**[i] can be performed in $\mathcal{O}(1)$ time. Furthermore, each vertex of T , except the root, is considered exactly once in both the for-loops spanning Steps 1–6 and 7–21. Each operation in these for-loops can be performed in $\mathcal{O}(1)$ time. Finally, the root is considered three times in Steps 22–28 and each operation in these three steps can be performed in $\mathcal{O}(1)$ time.

A total of $7n$ memory units are required to store the array **Labels**, while n memory units are required to store each of the five arrays **Parent**, **Branch**, **Previous1Label**, **A3Label**, **X** and the output of the function **EffectiveLabel**. Finally, three memory units are required to store the values of the variables i , ℓ and **prev**. The space complexity of Algorithm 5.6 is therefore $13n + 3 = \mathcal{O}(n)$. ■

5.2.4 Validation of the algorithm

The algorithm presented in §5.2.3 was implemented in Wolfram's **Mathematica** [106] and validated in five fundamentally different ways. First, the output of the algorithm was manually examined for a large number of randomly generated trees of varying orders, and it was confirmed in each case that the set X returned by the algorithm indeed represented a secure dominating set. Secondly, it was confirmed that the algorithm yielded minimum secure dominating sets for a number of trees T with special structure for which the value of $\gamma_s(T)$ is known (such as paths, stars, double stars and complete binary trees). Thirdly, it was verified that the value of $\gamma_s(T)$ returned by the algorithm did not violate any of the known theoretical bounds on the secure domination number of a graph, such as the bound

$$\frac{n}{1 + \Delta} \leq \gamma_s(G) \leq n - \nu,$$

for any (connected) graph G of order n with maximum degree Δ and matching number ν .

In addition, it was verified that the algorithm yields the same value of the secure domination number $\gamma_s(T)$ for a tree T when choosing different branch vertices as the root of T . This was done for randomly generated trees of different orders, choosing each branch vertex of each tree in turn as the root.

Trees of order	10	15	20	25	30
Algorithm 5.6 in §5.2.3	$\ll 0.01$	$\ll 0.01$	$\ll 0.01$	< 0.01	< 0.01
Branch-and-reduce algorithm in §5.1.1	0.05	0.93	14.16	263.54	2719.93
Branch-and-bound algorithm in §5.1.2	0.06	0.97	13.66	262.38	2692.03

TABLE 5.5: Comparison of the execution times of three algorithms when computing the secure domination numbers of trees of small order. Times are measured in seconds and are the averages of the times required for 30 randomly generated instances of trees of each order. All three algorithms were implemented in Wolfram's **Mathematica** [106] on a 3.4 GHz Intel(R) Core(TM) i7-3770 processor with 8 GiB RAM running in Ubuntu 12.04 [96].

Finally, it was verified that the algorithm yielded the same value of $\gamma_s(T)$ for trees T of small order as did the two exact algorithms (the branch-and-bound algorithm and the branch-and-reduce algorithm) for arbitrary graphs in §5.1. This was done for thirty randomly generated trees of orders 10, 15, 20, 25 and 30 each, noting the execution times of the three algorithms. These times are shown in Table 5.5. Although a much faster implementation of the algorithm in §5.2.3 possible when using a low-level programming language such as C or C++ instead of the high-level language **Mathematica**, the times in Table 5.5 serve the purpose of estimating the benefit (in the form of a speed-up factor) of using a linear, purpose-designed algorithm for computing the secure domination number of a tree rather than an exponential algorithm for general graphs.

5.3 Chapter summary

In this chapter, four algorithmic approaches towards finding minimum secure dominating sets of graphs were presented. The chapter opened in §5.1 with three algorithmic approaches towards finding a minimum secure dominating set of an arbitrary graph. Three rules were established in §4.1 for testing whether a subset of vertices of a graph is a secure dominating set. These three rules suffice by Theorem 3.6 and were ordered in increasing order of complexity thereby

representing an effective set of secure domination testing criteria for an arbitrary graph. The algorithms in §5.1 are based on these rules.

A branch-and-reduce algorithm for finding the secure domination number of an arbitrary graph was presented in §5.1.1. This algorithm is a recursive, depth-first tree search algorithm. It follows from Theorem 5.1 that the set S' of all support vertices of a graph G may, without loss of generality, be considered to form part of a minimum secure dominating set X of G , while set R' of all but one vertex from each redundancy class of G may be excluded from X , again without loss of generality. For any graph G , a branching decision is taken by the algorithm for each vertex $v \in V(G) - S' - R'$ in turn, namely whether to include v in X , or to exclude v from X . If v is included in X , the vertex set of G has to be repartitioned according to the description in §4.1 and a smaller instance of the problem may then be solved. In the latter case, a smaller instance of the problem is solved without having to repartition the vertex set of G . An external initialisation procedures produced the root of the search tree.

The branch-and-bound algorithm in §5.1.2 uses the same manner of partitioning the vertex set of G as in the branch-and-reduce approach. The branch-and-bound algorithm differs from the branch-and-reduce approach, because an upperbound and a lowerbound on $\gamma_s(G)$ are continually updated to bound the search tree and hence speed up the process of identifying a minimum secure dominating set of G . The worst-case time complexities for both the branch-and-reduce algorithm and the branch-and-bound algorithm were shown to be $\mathcal{O}(2^{n-s-\sum_{i=1}^k(|R_i|-1)})$ for a graph of order n with s support vertices and R_1, \dots, R_k redundancy classes. Numerical results obtained by the two algorithms were presented for grid graphs in the plane, grid graphs on the torus and circulant graphs.

A binary programming formulation was also presented for finding a minimum secure dominating set of an arbitrary graph. This formulation was tested for square grid graphs and square hexagonal graphs.

A linear algorithm was finally presented for finding a minimum secure dominating set of an arbitrary tree. After establishing a preliminary property of any secure dominating set of a tree in Lemma 5.1, a method was presented for constructing a minimum secure dominating set for a spider in §5.2.1. The algorithmic approach for finding a minimum secure domination set of an arbitrary tree T , presented in §5.2.2, entails including the vertices required in a minimum secure dominating set of a pendent spider \mathcal{S} of T , pruning away \mathcal{S} from T to form a smaller tree T' and repeating this process until only a final spider remains. It was shown in §5.2.3 that this algorithmic approach may be implemented in linear space and time. The various validation mechanisms applied to the algorithm were described in §5.2.4.

This section is concluded with a brief appraisal of the relative performances of the four algorithmic approaches described above for finding minimum secure dominating sets of graphs. The branch-and-bound algorithm slightly outperforms the branch-and-reduce algorithm for arbitrary graphs of small order. However, the branch-and-bound algorithm and the branch-and-reduce algorithm perform similarly for trees. The linear algorithm for trees can find the secure domination number of an arbitrary tree in a fraction of the time required by the branch-and-reduce algorithm or the branch-and-bound algorithm. The binary programming approach outperforms both the branch-and-reduce algorithm and the branch-and-bound algorithm for arbitrary graphs. It is estimated that even if the branch-and-reduce and the branch-and-bound algorithms were to be implemented in a low-level language, such as C or C++, these algorithms would still be outperformed by the binary programming implementation for arbitrary graphs. This may be attributed to the sophistication of the combination of exact and heuristic optimisation techniques employed by CPLEX 12.05, which is a state-of-the-art combinatorial optimisation tool. The

binary programming approach is, however, outperformed by the linear algorithm when trees are considered.

CHAPTER 6

Edge failure and secure graph domination

Contents

6.1	Two functions measuring the cost of edge failures	111
6.2	General bounds on the cost functions c_q and C_q	113
6.3	The cost functions c_q and C_q for paths and cycles	117
6.4	The cost functions c_q and C_q for wheels	122
6.5	The cost functions c_q and C_q for complete bipartite graphs	126
6.6	The cost functions c_q and C_q for complete graphs	131
6.7	Chapter summary	134

This chapter is concerned with determining the minimum and maximum increase in the secure domination number of a graph when edges are removed from the graph. The chapter opens in §6.1 with some basic definitions and results on two novel cost functions c_q and C_q that measure respectively the minimum and maximum increase in the secure domination number of a graph of size m when $q \in \{0, 1, \dots, m\}$ edges are removed from the graph. General bounds on the secure domination number of a graph are reviewed and established in §6.2, and these bounds are used to derive bounds on the two cost functions c_q and C_q . The remainder of the chapter is dedicated to determining the exact values of or tight bounds on the cost functions c_q and C_q for various special graph classes.

6.1 Two functions measuring the cost of edge failures

In applications, such as those mentioned in §1.2, the notion of edge failure is often important, because one might seek the cost (in terms of the additional number of guards required to protect a network of facilities in the secure dominating sense) if a number of edges of G fail (*i.e.* a number of links are eliminated from the network so that guards may no longer move along such disabled links).

The notation $G - qe$ is used to denote the set of all possible non-isomorphic subgraphs that can be obtained by removing $q \in \{0, 1, \dots, m\}$ edges from a graph G of size m . The notation $\gamma_s(G - qe)$ is similarly used to denote the set of values assumed by the secure domination number for elements of the set $G - qe$. A distinction is made between the graph obtained by removing a specific edge e from a given graph G , by writing $G - e$, and the class of graphs obtained by removing any single edge from G by writing $G - 1e$.

The effects of edge removal from a graph on the value of the secure domination number of the resulting graph are considered first.

Theorem 6.1 *Let G be any nonempty graph and let e be any edge of G . Then exactly one of the following statements is true:*

- (a) $\gamma_s(G - e) = \gamma_s(G)$, or
- (b) $\gamma_s(G - e) = \gamma_s(G) + 1$.

Proof: It follows from [12, Lemma 1], that $\gamma_s(G - e) \geq \gamma_s(G)$. Therefore, $\gamma_s(G - e) = \gamma_s(G) + k$ for some $k \in \mathbb{N}$. It is shown that $k \leq 1$. Let X be a minimum secure dominating set of G . If $e = uv$ joins two vertices in X , then clearly X is also a secure dominating set of $G - e$. Therefore, $\gamma_s(G - e) \leq |X| = \gamma_s(G)$ and so $\gamma_s(G - e) = \gamma_s(G)$. In this case, therefore, $k = 0$. Suppose next that e joins two vertices in $V(G) - X$. Then it follows by Theorem 3.6 that $\gamma_s(G - uv) = \gamma_s(G)$ if $u, v \notin X$ and u and v are not external private neighbours of X , and hence $k = 0$ again. If, however, $u, v \in \text{Epn}(x, X)$ for some $x \in X$, then $X \cup \{v\}$ is a secure dominating set of $G - e$ so that $\gamma_s(G - e) \leq |X \cup \{u\}| = \gamma_s(G) + 1$, in this case $k \leq 1$. If $u \in \text{Epn}(x_1, X)$ and $v \in \text{Epn}(x_2, X)$ for some $x_1, x_2 \in X$, then clearly X is also a secure dominating set of $G - e$. Therefore, $\gamma_s(G - e) \leq |X| = \gamma_s(G)$ and so $\gamma_s(G - e) = \gamma_s(G)$. Finally, suppose e joins a vertex $v \in X$ and a vertex $u \in V(G) - X$. Then $X \cup \{u\}$ is a secure dominating set of $G - e$ so that $\gamma_s(G - e) \leq |X \cup \{u\}| = \gamma_s(G) + 1$. In this final case, therefore, $k \leq 1$. ■

The following result is an immediate consequence of Theorem 6.1.

Corollary 6.1 (Edge removal cannot decrease the secure domination number)

For any graph G that is not edgeless, $\gamma_s(G) \leq \min \gamma_s(G - e) \leq \max \gamma_s(G - e) \leq \gamma_s(G) + 1$. ■

The cost functions

$$\begin{aligned} c_q(G) &= \min \gamma_s(G - qe) - \gamma_s(G) \\ \text{and} \\ C_q(G) &= \max \gamma_s(G - qe) - \gamma_s(G) \end{aligned}$$

are therefore non-negative in view of Corollary 6.1 and measure respectively the *smallest possible* and the *largest possible* increase in the minimum number of guards required to dominate an element of the set $G - qe$ securely, over and above the minimum number of guards required to dominate G securely, in the event that an arbitrary set of $q \in \{0, 1, \dots, m\}$ edges are removed from a graph G of size m .

The following inequalities follow from repeated application of Corollary 6.1.

Theorem 6.2 (Cost function q -growth properties)

If G is a graph of size $m > 0$ and $q \in \{0, 1, \dots, m\}$, then

- (a) $c_q(G) \leq c_{q+1}(G) \leq c_q(G) + 1$, and
- (b) $C_q(G) \leq C_{q+1}(G) \leq C_q(G) + 1$.

Proof: (a) By applying the result of Corollary 6.1, it follows that

$$\begin{aligned} c_{q+1}(G) &= \min \{ \gamma_s(G - (q+1)e) \} - \gamma_s(G) \\ &= \min \{ \gamma_s((G - qe) - 1e) \} - \gamma_s(G) \\ &\geq \min \{ \gamma_s(G - qe) \} - \gamma_s(G) \\ &= c_q(G), \end{aligned}$$

which establishes the first inequality. The second inequality holds because the secure domination number of a graph cannot increase by more than 1 if a single edge is removed from the graph according to Theorem 6.1. The proof of part (b) is similar. ■

The sequences $\mathbf{c}(G) = c_0(G), c_1(G), c_2(G), \dots, c_m(G)$ and $\mathbf{C}(G) = C_0(G), C_1(G), C_2(G), \dots, C_m(G)$ of cost functions may each be thought of as step functions (with steps of unit size) for any graph G of size m according to Theorem 6.2. Consider the path \mathcal{P}_6 as an example. The cost sequences for this graph are $\mathbf{c}(\mathcal{P}_6) = 0, 0, 0, 1, 2, 3$ and $\mathbf{C}(\mathcal{P}_6) = 0, 1, 1, 2, 2, 3$, as may be seen in Table 6.1.

q	$\mathcal{P}_6 - qe$	γ_s	$c_q(\mathcal{P}_6)$	$C_q(\mathcal{P}_6)$	Graphical representation
0	\mathcal{P}_6	3	0	0	
1	$\mathcal{P}_1 \cup \mathcal{P}_5$	4	0	1	
	$\mathcal{P}_2 \cup \mathcal{P}_4$	3			
2	$\mathcal{P}_3 \cup \mathcal{P}_3$	4	0	1	
	$2\mathcal{P}_1 \cup \mathcal{P}_4$	4			
3	$\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3$	4	1	2	
	$3\mathcal{P}_2$	3			
4	$3\mathcal{P}_1 \cup \mathcal{P}_3$	5	2	2	
	$2\mathcal{P}_1 \cup 2\mathcal{P}_2$	4			
5	$4\mathcal{P}_1 \cup \mathcal{P}_2$	5	2	2	
5	$6\mathcal{P}_1$	6	3	3	

TABLE 6.1: The cost functions $c_q(\mathcal{P}_6)$ and $C_q(\mathcal{P}_6)$ for the path \mathcal{P}_6 . Minimum secure dominating sets are denoted by solid vertices in the graphical representations.

6.2 General bounds on the cost functions c_q and C_q

General bounds on the secure domination number of a graph may be used to establish general bounds on the cost functions c_q and C_q .

Theorem 6.3 For any graph G of order n and size m with matching number ν ,

$$n - m \leq \gamma_s(G) \leq n - \nu.$$

Proof: It follows from Theorems 2.18 and 3.4 that

$$\gamma_s(G) \geq \gamma(G) \geq n - m \tag{6.1}$$

for any graph G of order n and size m , which establishes the lower bound.

Furthermore, a single vertex from each component of a matching of G securely dominates the matching, while the remaining vertices of G securely dominate themselves. Therefore,

$$\gamma_s(G) \leq \nu + (n - 2\nu) = n - \nu. \quad \blacksquare$$

The following result immediately follows from Corollary 6.1 and Theorem 6.3.

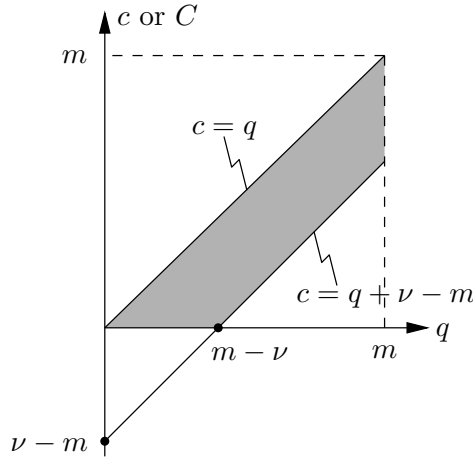


FIGURE 6.1: A graphical illustration of the bounds on $c_q(G)$ and $C_q(G)$ in Corollary 6.2. The graphs of the sequences of cost functions $\mathbf{c}(G)$ and $\mathbf{C}(G)$ are each step-functions of q which lie entirely within the shaded region.

Corollary 6.2 For any graph of order n and size m with matching number ν ,

$$\nu - m + q \leq c_q(G) \leq C_q(G) \leq q.$$

Proof: It follows from Theorem 6.3 that

$$\begin{aligned} c_q(G) &= \min\{\gamma_s(G - qe)\} - \gamma_s(G) \\ &\geq n - (m - q) - (n - \nu) \\ &= \nu - m + q. \end{aligned}$$

Furthermore, the secure domination number of G cannot increase by more than 1 if a single edge is removed from G by Corollary 6.1. By repeating this argument q times, it follows that $C_q(G) \leq q$, which completes the proof. ■

The bounds in Theorem 6.3 and Corollary 6.2 are sharp; they are attained by vertex disjoint unions of paths of orders 1 and 2. A graphical representation of the bounds in Corollary 6.2 may be found in Figure 6.1.

The lower bounds in Theorem 6.3 and Corollary 6.2 may, however, be improved slightly for connected graphs.

Corollary 6.3 If G is a connected graph of order n and size m other than \mathcal{K}_1 or \mathcal{K}_2 , then $\gamma_s(G) \geq n - m + 1$ and $c_q(G) \geq \nu - m + q + 1$.

Proof: The second inequality in (6.1) is strict if and only if G is a star. However, the first inequality in (6.1) is not strict if G is a star, except if $G \cong \mathcal{K}_1$ or $G \cong \mathcal{K}_2$. ■

The upper bound in Theorem 6.3 may be improved as a consequence of the following intermediate result.

Theorem 6.4 For any graph G , $\gamma_s(G) \leq \chi(\overline{G})$.

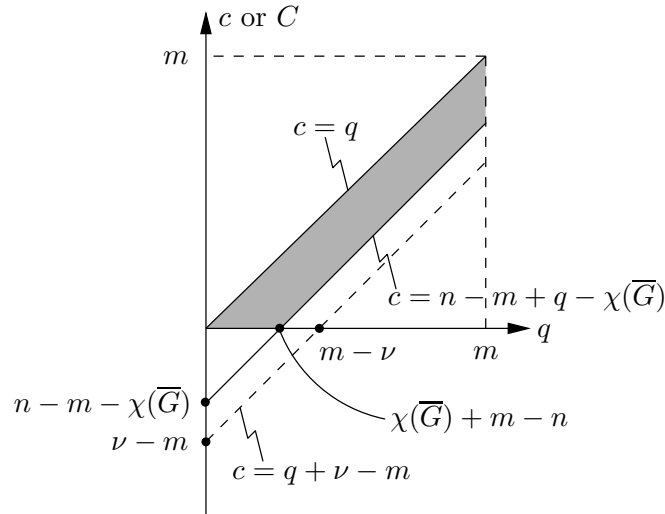


FIGURE 6.2: A graphical illustration of the bounds on $c_q(G)$ and $C_q(G)$ in Corollary 6.4. The graphs of $c(G)$ and $C(G)$ are step-functions of q which lie entirely within the shaded region.

Proof: Every colour class in a proper colouring of \overline{G} is an independent set in \overline{G} and hence induces a clique in G , which can be defended by a single vertex in that clique. ■

The next result follows immediately from Theorem 6.4.

Corollary 6.4 For any graph G of order n and size m , $c_q(G) \geq n - m + q - \chi(\overline{G})$.

Proof: It follows from Theorems 6.3 and 6.4 that

$$\begin{aligned} c_q(G) &= \min\{\gamma_s(G - qe)\} - \gamma_s(G) \\ &\geq n - (m - q) - \chi(\overline{G}) \\ &= n - m + q - \chi(\overline{G}). \end{aligned}$$

■

A graphical representation of the bound in Corollary 6.4 may be found in Figure 6.2. This lower bound is an improvement on the lower bound in Corollary 6.2, also shown in the figure.

Suppose the matching number of a graph G of order n is ν , and that G^* is a matching of G of size ν . Since each component of G^* forms an independent set in \overline{G} , a proper colouring of \overline{G} may be formed by assigning the same colour to both vertices in each component of G^* , but using a different colour for each component of G^* and by assigning all the vertices remaining in $G - G^*$ different colours. This shows that

$$\chi(\overline{G}) \leq \nu + (n - 2\nu) = n - \nu$$

and hence the upper bound on $\gamma_s(G)$ in Theorem 6.4 is better than the upper bound in Theorem 6.3 (and Corollary 6.2). Although the decision problem associated with computing $\chi(\overline{G})$ is **NP**-complete for general graphs, upper bounds on the chromatic number of a graph are therefore also upper bounds on $\gamma_s(G)$ in view of Theorem 6.4.

Theorem 6.5 For any graph G of order n with minimum degree δ , $\gamma_s(G) \leq n - \delta$.

Proof: It follows from Theorem 6.4 and Brooks' Theorem (see Theorem 2.2) that

$$\gamma_s(G) \leq \chi(\overline{G}) \leq \Delta(\overline{G}) = n - \delta$$

for any graph G other than a complete graph or an odd cycle. This result, in fact, also holds for complete graphs and odd cycles. For complete graphs it holds that

$$n - \delta(\mathcal{K}_n) = n - (n - 1) = 1 = \gamma_s(\mathcal{K}_n).$$

Similarly, for odd cycles ($n \geq 5$) it holds that

$$n - \delta(\mathcal{C}_n) = n - 2 \geq \left\lceil \frac{3n}{7} \right\rceil = \gamma_s(\mathcal{C}_n). \quad \blacksquare$$

The bound in Theorem 6.5 is sharp; it is attained by a star.

Corollary 6.5 *For any graph G of order n and size m with minimum degree δ ,*

$$c_q(G) \geq \delta - m + q.$$

Proof: It follows from Theorems 6.3 and 6.5 that

$$\begin{aligned} c_q(G) &= \min\{\gamma_s(G - qe)\} - \gamma_s(G) \\ &\geq n - (m - q) - (n - \delta) \\ &= \delta - m + q. \end{aligned} \quad \blacksquare$$

The next result shows that the secure domination of an arbitrary graph is bounded from above by the 2-tuple domination number of a graph.

Theorem 6.6 *For any graph G , $\gamma_s(G) \leq \gamma_{\times 2}(G)$.*

Proof: Suppose X is a minimum 2-tuple dominating set of G . Then X is also a dominating set of G . Let $v \in X$, but suppose $u \notin X$ with $u \in N(v) \cap (V(G) - X)$. The swap set $(X - \{v\}) \cup \{u\}$ is again a dominating set of G , since the vertices $N(v) \cap (V(G) - X)$ are dominated by at least one vertex other than v , while all vertices $(V(G) - X) - \{u\}$ are dominated by at least two vertices in X . \blacksquare

The next result is an immediate consequence of Theorem 6.6.

Corollary 6.6 *For any graph G of order n and size m ,*

$$c_q(G) \geq n - m + q - \gamma_{\times 2}(G).$$

Proof: It follows from Theorems 6.3 and 6.6 that

$$\begin{aligned} c_q(G) &= \min\{\gamma_s(G - qe)\} - \gamma_s(G) \\ &\geq n - (m - q) - \gamma_{\times 2}(G) \\ &= n - m + q - \gamma_{\times 2}(G). \end{aligned} \quad \blacksquare$$

It is also possible to establish a lower bound on $c_q(G)$ in terms of both the order and size of G .

Theorem 6.7 For any graph G of order n and size m ,

$$\gamma_s(G) \leq \frac{1}{2} \left(1 + \sqrt{4n(n-1) - 8m + 1} \right).$$

Proof: It follows from Theorems 6.4 and 2.3 that

$$\begin{aligned} \gamma_s(G) &\leq \chi(\overline{G}) \\ &\leq \frac{1}{2} \left(1 + \sqrt{8m(\overline{G}) + 1} \right) \\ &= \frac{1}{2} \left(1 + \sqrt{8 \left(\binom{n}{2} - m \right) + 1} \right) \\ &= \frac{1}{2} \left(1 + \sqrt{4n(n-1) - 8m + 1} \right). \end{aligned}$$

■

The next bound follows immediately due to Theorem 6.7.

Corollary 6.7 For any graph G of order n and size m ,

$$c_q(G) \geq n - m + q - \frac{1}{2} \left(1 + \sqrt{4n(n-1) - 8m + 1} \right).$$

Proof: It follows from Theorems 6.3 and 6.7 that

$$\begin{aligned} c_q(G) &= \min\{\gamma_s(G - qe)\} - \gamma_s(G) \\ &\geq n - (m - q) - \frac{1}{2} \left(1 + \sqrt{4n(n-1) - 8m + 1} \right) \\ &= n - m + q - \frac{1}{2} \left(1 + \sqrt{4n(n-1) - 8m + 1} \right). \end{aligned}$$

■

6.3 The cost functions c_q and C_q for paths and cycles

It follows by Corollaries 6.2 and 6.4 that

$$1 + q - \left\lceil \frac{n}{2} \right\rceil \leq c_q(\mathcal{P}_n) \leq C_q(\mathcal{P}_n) \leq q$$

for a path \mathcal{P}_n of order $n \geq 2$ and any $q \in \{0, 1, \dots, m\}$, by noting that $\chi(\overline{\mathcal{P}_n}) = \lceil \frac{n}{2} \rceil$. However, these bounds are weak, especially for small values of q . The cost function sequences $\mathbf{c}(\mathcal{P}_n)$ and $\mathbf{C}(\mathcal{P}_n)$ are determined exactly in this section, and this is followed by a derivation of the sequences $\mathbf{c}(\mathcal{C}_n)$ and $\mathbf{C}(\mathcal{C}_n)$ for a cycle \mathcal{C}_n from these results. The following basic result is required.

Lemma 6.1

- (a) For $n \geq 8$ and any $1 \leq k < n$, $\gamma_s(\mathcal{P}_k \cup \mathcal{P}_{n-k}) \geq \gamma_s(\mathcal{P}_7 \cup \mathcal{P}_{n-7})$.
- (b) For $n \geq 6$ and any $1 \leq k < n$, $\gamma_s(\mathcal{P}_k \cup \mathcal{P}_{n-k}) \leq \gamma_s(\mathcal{P}_5 \cup \mathcal{P}_{n-5})$.

Proof: (a) Suppose $n \geq 8$ and let k be any positive integer not exceeding $n - 1$. Then

$$\begin{aligned} \gamma_s(\mathcal{P}_k) + \gamma_s(\mathcal{P}_{n-k}) &= \left\lceil \frac{3(k)}{7} \right\rceil + \left\lceil \frac{3(n-k)}{7} \right\rceil \\ &\geq \left\lceil \frac{3n}{7} \right\rceil \\ &= 3 + \left\lceil \frac{3n}{7} - 3 \right\rceil \\ &= \left\lceil \frac{3(7)}{7} \right\rceil + \left\lceil \frac{3(n-7)}{7} \right\rceil \\ &= \gamma_s(\mathcal{P}_7) + \gamma_s(\mathcal{P}_{n-7}) \end{aligned}$$

by the identity $\lceil a \rceil + \lceil b - a \rceil \geq \lceil b \rceil$ for any $a, b \in \mathbb{R}$ (see Proposition A.1 in Appendix B).

(b) Suppose $n \geq 6$ and let k be any positive integer not exceeding $n - 1$. Then

$$\begin{aligned} \gamma_s(\mathcal{P}_k) + \gamma_s(\mathcal{P}_{n-k}) &= \left\lceil \frac{3(k)}{7} \right\rceil + \left\lceil \frac{3(n-k)}{7} \right\rceil \\ &= \left\lceil \frac{3(k)}{7} + \frac{6}{7} \right\rceil + \left\lceil \frac{3(n-k)}{7} + \frac{6}{7} \right\rceil \\ &\leq \left\lceil \frac{3n}{7} + \frac{6}{7} + \frac{6}{7} \right\rceil \\ &= \left\lceil \frac{3n}{7} + \frac{6}{7} \right\rceil \\ &= \left\lceil \frac{3(5)}{7} \right\rceil + \left\lceil \frac{3(n-5)}{7} \right\rceil \\ &= \gamma_s(\mathcal{P}_5 \cup \mathcal{P}_{n-5}) \end{aligned}$$

by (three times) using the identity $\lceil \frac{a}{b} \rceil = \left\lfloor \frac{a+b-1}{b} \right\rfloor = \left\lfloor \frac{a}{b} + \frac{b-1}{b} \right\rfloor$ for any $a, b \in \mathbb{R}$ with $b \neq 0$ (see Proposition A.5 in Appendix B). ■

The following intermediate results are also necessary for determining the sequences $\mathbf{c}(\mathcal{P}_n)$ and $\mathbf{C}(\mathcal{P}_n)$ exactly.

Lemma 6.2 Suppose $E, F \in \mathcal{P}_n - qe$ respectively minimise and maximise $\gamma_s(\mathcal{P}_n - qe)$.

- (a) If $2q \leq n \leq 2q + 4$, then $E \cup \mathcal{P}_2$ minimises $\gamma_s(\mathcal{P}_{n+2} - (q+1)e)$.
- (b) If $2q \leq n$, then $E \cup \mathcal{P}_7$ minimises $\gamma_s(\mathcal{P}_{n+7} - (q+1)e)$.
- (c) If $n - 4 \leq q \leq n - 1$, then $F \cup \mathcal{P}_1$ maximises $\gamma_s(\mathcal{P}_{n+1} - (q+1)e)$.
- (d) If $q \leq n - 1$, then $F \cup \mathcal{P}_5$ maximises $\gamma_s(\mathcal{P}_{n+5} - (q+1)e)$.

Proof: (a) By contradiction. Suppose, to the contrary, that $2q \leq n \leq 2q + 4$ and that $G \in \mathcal{P}_{n+2} - (q+1)e$ minimises $\gamma_s(\mathcal{P}_{n+2} - (q+1)e)$, but that $\gamma_s(G) < \gamma_s(E \cup \mathcal{P}_2)$. Then G contains no component isomorphic to \mathcal{P}_2 . It is shown next that it may be assumed that G is isolate-free. Since $\gamma_s(\mathcal{P}_i) \leq \gamma_s(\mathcal{P}_{i+1})$ for all $i \in \mathbb{N}$, it follows that $\gamma_s(\mathcal{P}_2 \cup \mathcal{P}_\ell) \leq \gamma_s(\mathcal{P}_1 \cup \mathcal{P}_{\ell+1})$. This means that if G were to contain a component of order 1, then G would have no component of order $i \geq 2$. But if G is the empty graph of order $n + 2$, then $q = n + 1$, which contradicts the supposition that $n \geq 2q$. Furthermore, G can have at most one component of order 3, since $\gamma_s(\mathcal{P}_3 \cup \mathcal{P}_3) > \gamma_s(\mathcal{P}_4 \cup \mathcal{P}_2)$. But then the order of G is $n + 2 > 3(q + 2)$, which contradicts the supposition that $n \leq 2q + 4$.

(b) By contradiction. Suppose, to the contrary, that $2q \leq n$ and that $G \in \mathcal{P}_{n+7} - (q+1)e$ minimises $\gamma_s(\mathcal{P}_{n+7} - (q+1)e)$, but that $\gamma_s(G) < \gamma_s(E \cup \mathcal{P}_7)$. Then G contains no component of order 7 and it follows by Lemma 6.1(a) that no two components of G together have more than seven vertices. Furthermore, the inequality $\gamma_s(2\mathcal{P}_3) = 4 > 3 = \gamma_s(\mathcal{P}_2 \cup \mathcal{P}_4)$ and the equality $\gamma_s(\mathcal{P}_3 \cup \mathcal{P}_4) = 4 = \gamma_s(\mathcal{P}_2 \cup \mathcal{P}_5) = \gamma_s(\mathcal{P}_1 \cup \mathcal{P}_6)$ show that there is at least one member of $\mathcal{P}_{n+7} - (q+1)e$ which minimises $\gamma_s(\mathcal{P}_{n+7} - (q+1)e)$ and which has at most one component

not isomorphic to \mathcal{P}_1 or \mathcal{P}_2 . It may therefore be assumed that $G \cong \mathcal{P}_i \cup x\mathcal{P}_2 \cup y\mathcal{P}_1$ for some $i \in \{2, 3, 4, 5, 6\}$. By evaluating the number of components and the number of vertices of G , it follows that $1 + x + y = q + 2$ and $i + 2x + y = n + 7$, respectively. The unique solution to this simultaneous system of equations is $x = n - q - i + 6$ and $y = 2q + i - 5 - n$. But since $y \geq 0$, it follows that $n \leq 2q + i - 5$, which contradicts the supposition that $2q \leq n$ for $i = 2, 3, 4$. Furthermore, $i \neq 5$, because $\gamma_s(\mathcal{P}_2 \cup \mathcal{P}_4) = 3 < 4 = \gamma_s(\mathcal{P}_1 \cup \mathcal{P}_5)$. Finally, if $i = 6$, then $x = 0$ in order to avoid contradicting Lemma 6.1(a). But $i = 6$ and $x = 0$ implies that $n = q$, again contradicting the supposition. It is concluded, therefore, that $G \cong x\mathcal{P}_2 \cup y\mathcal{P}_1$. But in this case it follows by a similar argument as the one above that $x = n - q + 5$ and $y = 2q - n - 3$. Since $y \geq 0$ it follows that $2q \geq n - 3$, yet again contradicting the supposition.

(c) By contradiction. Suppose, to the contrary, that $n - 4 \leq q \leq n - 1$ and that $H \in \mathcal{P}_{n+1} - (q+1)e$ maximises $\gamma_s(\mathcal{P}_{n+1} - (q+1)e)$, but that $\gamma_s(H) > \gamma_s(F \cup \mathcal{P}_1)$. Then H is isolate-free and has at most one component of order 2, because $\gamma_s(\mathcal{P}_2 \cup \mathcal{P}_2) = 2 < 3 = \gamma_s(\mathcal{P}_3 \cup \mathcal{P}_1)$. But then the order of H is $n + 1 > 2(q + 2)$, which contradicts the supposition that $n \leq q + 4$.

(d) By contradiction. Suppose, to the contrary, that $q \leq n - 1$ and that $H \in \mathcal{P}_{n+5} - (q+1)e$ maximises $\gamma_s(\mathcal{P}_{n+5} - (q+1)e)$, but that $\gamma_s(H) > \gamma_s(F \cup \mathcal{P}_5)$. Then H contains no component of order 5 and it follows by Lemma 6.1(b) that no two components of H together have more than five vertices. Furthermore, the inequality $\gamma_s(\mathcal{P}_2 \cup \mathcal{P}_2) = 2 < 3 = \gamma_s(\mathcal{P}_3 \cup \mathcal{P}_1)$ and the equality $\gamma_s(\mathcal{P}_2 \cup \mathcal{P}_3) = 3 = \gamma_s(\mathcal{P}_4 \cup \mathcal{P}_1)$ show that there is at least one member of $\mathcal{P}_{n+5} - (q+1)e$ which maximises $\gamma_s(\mathcal{P}_{n+5} - (q+1)e)$ and which has at most one component that is not an isolate. It may therefore be assumed that $G \cong \mathcal{P}_i \cup x\mathcal{P}_1$ for some $i \in \{2, 3, 4\}$. By evaluating the number of components and the number of vertices of H , it follows that $x + 1 = q + 2$ and $x + i = n + 5$, respectively, which together imply that $n = q + i - 4$. But this equality contradicts the supposition that $q \leq n - 1$ for $i = 2, 3, 4$. ■

The sequences of cost functions \mathbf{c} and \mathbf{C} may now be established for paths.

Theorem 6.8 (The sequences \mathbf{c} and \mathbf{C} for paths)

For any $n \in \mathbb{N}$ and any nonnegative integer $q \leq n - 1$,

$$c_q(\mathcal{P}_n) = \begin{cases} 0 & \text{if } q < \frac{n}{7} \\ \left\lceil \frac{2n+q+1}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil & \text{if } \frac{n}{7} \leq q \leq \frac{n}{2} \\ q + 1 - \left\lceil \frac{3n}{7} \right\rceil & \text{if } q > \frac{n}{2} \end{cases} \quad \text{and} \quad C_q(\mathcal{P}_n) = \begin{cases} \left\lceil \frac{3n+6q}{7} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil & \text{if } q < \frac{n}{5} \\ \left\lceil \frac{n+q}{2} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil & \text{if } q \geq \frac{n}{5}. \end{cases}$$

Proof: All three cases of the formula above for $c_q(\mathcal{P}_n)$ are established by induction over q . Suppose $n > 7q$, for which the base case is $c_0(\mathcal{P}_n) = 0$ and that $E_n \in \mathcal{P}_n - \ell e$ minimises $\gamma_s(\mathcal{P}_n - \ell e)$. Assume, as induction hypothesis, that the desired formula holds for $q = \ell$, i.e. $\min\{\gamma_s(\mathcal{P}_n - \ell e)\} = \left\lceil \frac{3n}{7} \right\rceil$ for all $\ell < \frac{n}{7}$. To show that the formula also holds for $q = \ell + 1$, a disjoint path \mathcal{P}_7 is added to E_n for all $n > 7\ell$. Then it follows by Lemma 6.2(b) that

$$\min\{\gamma_s(\mathcal{P}_{n+7} - (\ell + 1)e)\} = \min\{\gamma_s(\mathcal{P}_n - \ell e)\} + \gamma_s(\mathcal{P}_7) = \left\lceil \frac{3n}{7} \right\rceil + 3 = \left\lceil \frac{3(n+7)}{7} \right\rceil,$$

showing that $c_{\ell+1}(\mathcal{P}_{n+7}) = 0$ for all $n > 7(\ell + 1)$ and thereby completing the induction process for this case.

Suppose next that $2q \leq n \leq 7q$. It may easily be verified that the base case holds for $2 \leq n \leq 7$ and $q = 1$. Suppose that $E_n \in \mathcal{P}_n - \ell e$ minimises $\gamma_s(\mathcal{P}_n - \ell e)$ and assume, as induction hypothesis, that the formula holds for $q = \ell$, i.e. $\min\{\gamma_s(\mathcal{P}_n - \ell e)\} = \left\lceil \frac{2n+\ell+1}{5} \right\rceil$ for all $2\ell \leq n \leq 7\ell$. To show

that the formula also holds for $q = \ell + 1$, a disjoint path \mathcal{P}_2 is added to E_n for $2\ell \leq n \leq 2\ell + 4$, while a disjoint path \mathcal{P}_7 is added to E_n for $2\ell + 5 \leq n \leq 7\ell$, thereby covering the required range of values of n for $q = \ell + 1$, *i.e.* $2\ell + 2 \leq n \leq 7\ell + 7$. Then it follows by Lemma 6.2(a) that

$$\begin{aligned} \min\{\gamma_s(\mathcal{P}_{n+2} - (\ell + 1)e)\} &= \min\{\gamma_s(\mathcal{P}_n - qe)\} + \gamma_s(\mathcal{P}_2) \\ &= \left\lceil \frac{2n + \ell + 1}{5} \right\rceil + 1 \\ &= \left\lceil \frac{2n + \ell + 1 + 5}{5} \right\rceil \\ &= \left\lceil \frac{2(n + 2) + (\ell + 1) + 1}{5} \right\rceil, \end{aligned}$$

thereby completing the induction process for $2\ell \leq n \leq 2\ell + 4$. Furthermore, it follows by Lemma 6.2(b) that

$$\begin{aligned} \min\{\gamma_s(\mathcal{P}_{n+7} - (\ell + 1)e)\} &= \min\{\gamma_s(\mathcal{P}_n - \ell e)\} + \gamma_s(\mathcal{P}_7) \\ &= \left\lceil \frac{2n + \ell + 1}{5} \right\rceil + 3 \\ &= \left\lceil \frac{2n + \ell + 1 + 15}{5} \right\rceil \\ &= \left\lceil \frac{2(n + 7) + (\ell + 1) + 1}{5} \right\rceil, \end{aligned}$$

thereby completing the induction process for $2\ell + 5 \leq n \leq 7\ell$.

Finally, suppose $n < 2q$ and consider $c_2(\mathcal{P}_3) = 1$ as base case. Assume, as induction hypothesis, that the formula holds for $q = \ell$, *i.e.* $\min\{\gamma_s(\mathcal{P}_n - \ell e)\} = q + 1$ for $n < 2\ell$. Let $E_n \in \mathcal{P}_n - \ell e$ and suppose the vertex set of E_n is $\{v_1, \dots, v_n\}$. It is shown by contradiction that E_n has at least one isolated vertex. Suppose, to the contrary, that E_n has no isolated vertex. Then it follows by the handshaking lemma (see Theorem 2.1) that

$$n \leq \sum_{i=1}^n \deg(v_i) = 2m = 2(n - 1 - \ell),$$

since each vertex has degree at least one. Therefore, $n \leq 2(n - 1 - \ell)$, or equivalently $n \geq 2\ell + 2$, which contradicts the fact that $n < 2\ell + 2$. Hence E_n has at least one isolated vertex, and so

$$\begin{aligned} \min\{\gamma_s(\mathcal{P}_{n+1} - (\ell + 1)e)\} &= \min\{\gamma_s(\mathcal{P}_n - \ell e)\} + \gamma_s(\mathcal{P}_1) \\ &= (\ell + 1) + 1, \end{aligned}$$

thereby completing the induction process.

Both cases of the formula above for $C_q(\mathcal{P}_n)$ are established by induction over q and suppose that $5q < n$ and suppose that $F_n \in \mathcal{P}_n - \ell e$ maximises $\gamma_s(\mathcal{P}_n - \ell e)$ and assume, as induction hypothesis, that the formula holds for $q = \ell$, *i.e.* $\max\{\gamma_s(\mathcal{P}_n - \ell e)\} = \left\lceil \frac{3n + 6\ell}{7} \right\rceil$ for all $5\ell < n$. To show that the formula also holds for $q = \ell + 1$, a disjoint path \mathcal{P}_5 is added to F_n for $5q < n$, thereby covering the required range of values of n for $q = \ell + 1$, *i.e.* $5\ell + 5 < n$. Then it follows

by Lemma 6.2(d) that

$$\begin{aligned}
 \max\{\gamma_s(\mathcal{P}_{n+1} - (\ell + 1)e)\} &= \max\{\gamma_s(\mathcal{P}_n - qe)\} + \gamma_s(\mathcal{P}_5) \\
 &= \left\lceil \frac{3n + 6\ell}{7} \right\rceil + 3 \\
 &= \left\lceil \frac{3n + 6\ell + 21}{7} \right\rceil \\
 &= \left\lceil \frac{3(n + 5) + 6(\ell + 1)}{5} \right\rceil,
 \end{aligned}$$

thereby completing the induction process for $5\ell < n$.

Suppose next that $n \leq 5q$ and suppose that $F_n \in \mathcal{P}_n - \ell e$ maximises $\gamma_s(\mathcal{P}_n - \ell e)$. Assume, as induction hypothesis, that the formula holds for $q = \ell$, i.e. $\max\{\gamma_s(\mathcal{P}_n - \ell e)\} = \lceil \frac{n+\ell}{2} \rceil$ for all $n \leq 5\ell$. To show that the formula also holds for $q = \ell + 1$, a disjoint path \mathcal{P}_1 is added to F_n for $n - 4 \leq \ell \leq n - 1$, while a disjoint path \mathcal{P}_5 is added to F_n for $\ell \leq n - 5$, thereby covering the required range of values of n for $q = \ell + 1$, i.e. $n \leq 5\ell + 5$. It follows by Lemma 6.2(d) that

$$\begin{aligned}
 \max\{\gamma_s(\mathcal{P}_{n+5} - (\ell + 1)e)\} &= \max\{\gamma_s(\mathcal{P}_n - qe)\} + \gamma_s(\mathcal{P}_5) \\
 &= \left\lceil \frac{n + \ell + 1}{2} \right\rceil + 3 \\
 &= \left\lceil \frac{n + \ell + 1 + 6}{2} \right\rceil \\
 &= \left\lceil \frac{(n + 5) + (\ell + 1) + 1}{5} \right\rceil,
 \end{aligned}$$

thereby completing the induction process for $n - 4 \leq \ell \leq n - 1$. Furthermore, it follows by Lemma 6.2(d) that

$$\begin{aligned}
 \max\{\gamma_s(\mathcal{P}_{n+1} - (\ell + 1)e)\} &= \max\{\gamma_s(\mathcal{P}_n - qe)\} + \gamma_s(\mathcal{P}_1) \\
 &= \left\lceil \frac{n + \ell + 1}{2} \right\rceil + 1 \\
 &= \left\lceil \frac{n + \ell + 1 + 2}{2} \right\rceil \\
 &= \left\lceil \frac{(n + 1) + (\ell + 1) + 1}{5} \right\rceil,
 \end{aligned}$$

thereby completing the induction process for $n \leq 5q$. ■

The next result follows immediately from Theorem 6.8, because $\mathcal{C}_n - 1e$ contains a single element, which is isomorphic to \mathcal{P}_n , for all $n \geq 3$.

Corollary 6.8 (The sequences c and C for cycles)

For any $n \in \mathbb{N}$ and any nonnegative integer $q \leq n$,

$$c_q(\mathcal{C}_n) = \begin{cases} 0 & \text{if } q < \frac{n}{7} + 1 \\ \left\lceil \frac{2n+q}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil & \text{if } \frac{n}{7} + 1 \leq q \leq \frac{n}{2} + 1, \\ q - \left\lceil \frac{3n}{7} \right\rceil & \text{if } q > \frac{n}{2} + 1. \end{cases}$$

and

$$C_q(\mathcal{C}_n) = \begin{cases} \left\lceil \frac{3n+6q-6}{7} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil & \text{if } q < \frac{n}{5} + 1 \\ \left\lceil \frac{n+q-1}{2} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil & \text{if } \frac{n}{5} + 1 \leq q \leq n. \end{cases}$$

6.4 The cost functions c_q and C_q for wheels

Klostermeyer and Mynhardt [68] determined the secure total domination number of a wheel. They showed that $\gamma_{st}(\mathcal{W}_n) = \lceil n/3 \rceil + 1$ for a wheel $\mathcal{W}_n = \mathcal{C}_n + v_0$ of order $n + 1$, containing $n \geq 4$ spokes. The hub vertex and every third vertex on \mathcal{C}_n forms a minimum secure total dominating set of \mathcal{W}_n [68]. The value of $\gamma_s(\mathcal{W}_n)$ is established in the following result.

Theorem 6.9 *For any wheel \mathcal{W}_n of order $n + 1$, $\gamma_s(\mathcal{W}_n) = \lceil \frac{n+1}{3} \rceil$.*

Proof: Suppose the vertex set of the n -cycle \mathcal{C}_n is $\{v_1, \dots, v_n\}$ and consider the wheel $\mathcal{W}_n = \mathcal{C}_n + v_0$. Let $n = 3k + r$, where $k \geq 1$ and $r \in \{0, 1, 2\}$, and define

$$Y = \bigcup_{j=1}^{k-1} \{v_{3j+1}\} \quad \text{and} \quad Z = \begin{cases} v_{3k} & \text{if } r = 0, 1 \\ v_{3k+1} & \text{if } r = 2. \end{cases}$$

Then $X = Y \cup Z \cup \{v_0\}$ is a secure dominating set of \mathcal{W}_n , implying that $\gamma_s(\mathcal{W}_n) \leq |X| = k + 1 = \lceil (n + 1)/3 \rceil$.

It is next shown by contradiction that X is a minimum secure dominating set of \mathcal{W}_n . Suppose, to the contrary, that X' is a secure dominating set of \mathcal{W}_n such that $|X'| < |X|$. It follows that $|X'| \leq \lceil (n - 2)/3 \rceil$.

First consider the case where $v_0 \notin X'$. In this case X' is not a dominating set of $V(\mathcal{W}_n) - \{v_0\}$ for $n \equiv 1, 2 \pmod{3}$, since $\lceil (n - 2)/3 \rceil < \gamma(\mathcal{W}_n - \{v_0\}) = \gamma(\mathcal{C}_n) = \lceil n/3 \rceil$, a contradiction. Furthermore, if $n \equiv 0 \pmod{3}$, then X' can only be a dominating set of \mathcal{W}_n if every third vertex in $V(\mathcal{C}_n)$ is in X' , but then no vertex in $V(\mathcal{C}_n) - X'$ is defended, again a contradiction.

Suppose, therefore, that $v_0 \in X'$. Then at least three consecutive vertices v_i, v_{i+1}, v_{i+2} for some $0 \leq i \leq n - 2$ from the set $V(\mathcal{C}_n) = \{v_1, \dots, v_n\}$ have the property that $X' \cap N(v_i) \cap N(v_{i+1}) \cap N(v_{i+2}) = \{v_0\}$. Since $G[\text{Epn}(v_0, X') \cup \{v_0\}]$ is not complete, X' is not a secure dominating set of \mathcal{W}_n , again a contradiction. ■

Using the result of Theorem 6.9, it is possible to find good bounds on $c(\mathcal{W}_n)$.

Theorem 6.10 *Suppose $k \in \mathbb{N}$. Then*

$$(a) \quad c_q(\mathcal{W}_{3k}) \leq \begin{cases} 0 & \text{if } 0 \leq q \leq 2k \\ \ell & \text{if } 2k + 3(\ell - 1) < q \leq 2k + 3\ell, \text{ for } \ell = 1, \dots, k \\ k + j & \text{if } 5k + j \leq q \leq 6k, \text{ for } j = 1, \dots, k, \end{cases}$$

$$(b) \quad c_q(\mathcal{W}_{3k+1}) \leq \begin{cases} 0 & \text{if } 0 \leq q \leq 2k + 1 \\ 1 & \text{if } 2k + 1 < q \leq 2k + 2 \\ \ell + 1 & \text{if } 2k + 3(\ell - 1) + 2 < q \leq 2k + 3\ell + 2, \text{ for } \ell = 1, \dots, k \\ k + 1 + j & \text{if } 5k + 2 + j \leq q \leq 6k + 2, \text{ for } j = 1, \dots, k, \end{cases}$$

$$(c) \quad c_q(\mathcal{W}_{3k+2}) \leq \begin{cases} 0 & \text{if } 0 \leq q \leq 2k + 1 \\ 1 & \text{if } 2k + 1 < q \leq 2k + 3 \\ \ell + 1 & \text{if } 2k + 3(\ell - 1) + 3 < q \leq 2k + 3\ell + 3, \text{ for } \ell = 1, \dots, k \\ k + 1 + j & \text{if } 5k + 3 + j \leq q \leq 6k + 4, \text{ for } j = 1, \dots, k + 1. \end{cases}$$

Proof: (a) Denote the hub of \mathcal{W}_{3k} by v_0 and the vertices of the remaining cycle \mathcal{C}_{3k} by v_1, \dots, v_{3k} . Then $X = \left(\bigcup_{j=1}^k \{v_{3j-1}\}\right) \cup \{v_0\}$ is a minimum secure dominating set of \mathcal{W}_{3k} by Theorem 6.9. The following three cases are considered in establishing the formula for $c_q(\mathcal{W}_{3k})$:

Case a(i): $0 \leq q \leq 2k$. The set of vertices X remains a secure dominating set upon the removal of any q edges from the set $\{v_3v_4, v_6v_7, v_9v_{10}, \dots, v_{3k}v_1\} \cup \{v_0v_2, v_0v_5, v_0v_8, \dots, v_0v_{3k-1}\}$, showing that $c_q(\mathcal{W}_{3k}) = 0$ in this case.

Case a(ii): $2k + 3(\ell - 1) < q \leq 2k + 3\ell$ for $\ell = 1, \dots, k$. Removal of the $2k$ edges described in *Case a(i)* yields a graph comprising k four-cycles $\{v_0, v_{3j-1}, v_{3j}, v_{3j+1}\}$ for $j = 1, \dots, k$ joined to the vertex v_0 . This graph admits X as minimum secure dominating set. By removing a further $3k$ edges in the order $v_0v_{3j-2}, v_0v_{3j}, v_{3j-1}v_{3j}$, starting with $j = 1, \dots, k$, the desired lower bound arises for $c_q(\mathcal{W}_{3k})$. In this case

$$X \cup \left(\bigcup_{j=1}^k \{v_{3j}\}\right)$$

is a minimum secure dominating set of the resulting graph.

Case a(iii): $5k + j \leq q \leq 6k$ for $j = 1, \dots, k$. Removal of the $5k$ edges as described in *Cases a(i)* and *a(ii)* above yields a graph isomorphic to $(k + 1)\mathcal{P}_1 \cup k\mathcal{P}_2$, for which

$$X \cup \left(\bigcup_{j=1}^k \{v_{3j}\}\right)$$

is a minimum secure dominating set. Upon removal of the remaining edges, in any order, the desired lower bound arises for $c_q(\mathcal{W}_{3k})$.

(b) Denote the hub of \mathcal{W}_{3k+1} by v_0 and the vertices of the remaining cycle \mathcal{C}_{3k+1} by v_1, \dots, v_{3k+1} . It follows that $X = \left(\bigcup_{j=1}^k \{v_{3j}\}\right) \cup \{v_0\}$ is a minimum secure dominating set of \mathcal{W}_{3k+1} by Theorem 6.9. The following four cases are considered in establishing the formula for $c_q(\mathcal{W}_{3k+1})$:

Case b(i): $0 \leq q \leq 2k + 1$. The set of vertices X remains a secure dominating set upon the removal of any q edges from the set $\{v_1v_2, v_4v_5, v_7v_8, \dots, v_{3k+1}v_1\} \cup \{v_0v_3, v_0v_6, v_0v_9, \dots, v_0v_{3k}\}$, showing that $c_q(\mathcal{W}_{3k+1}) = 0$ in this case.

Case b(ii): $2k + 1 < q \leq 2k + 2$. Removal of the $2k + 1$ edges described in *Case b(i)* yields a graph comprising k four-cycles $\{v_0, v_{3j-1}, v_{3j}, v_{3j+1}\}$ for $j = 1, \dots, k$ and a path $\{v_0, v_1\}$ which are all joined to the vertex v_0 . This graph admits X as secure dominating set. Upon the removal of the edge v_0v_1 , the set $X \cup \{v_1\}$ is a minimum secure dominating set.

Case b(iii): $2k + 3(\ell - 1) + 2 < q \leq 2k + 3\ell + 2$ for $\ell = 1, \dots, k$. Removal of the $2k + 2$ edges described in *Cases b(i)* and *b(ii)* yields a graph comprising k four-cycles $\{v_0, v_{3j-1}, v_{3j}, v_{3j+1}\}$ for $j = 1, \dots, k$ joined to the vertex v_0 and an isolated vertex v_1 . This graphs admits $X \cup \{v_1\}$ as secure dominating set. By removing a further $3k$ edges in the order $v_0v_{3j-1}, v_0v_{3j+1}, v_{3j}v_{3j+1}$, starting with $j = 1, \dots, k$, the desired lower bound arises for $c_q(\mathcal{W}_{3k+1})$. In this case

$$X \cup \{v_1\} \cup \left(\bigcup_{j=1}^k \{v_{3j+1}\}\right)$$

is a minimum secure dominating set of the resulting graph.

Case b(iv): $5k + 2 + j \leq q \leq 6k + 2$ for $j = 1, \dots, k$. Removal of the $5k + 2$ edges in *Cases b(i)–b(iii)* yields a graph isomorphic to $(k + 2)\mathcal{P}_1 \cup k\mathcal{P}_2$. This graph admits

$$X \cup \{v_1\} \cup \left(\bigcup_{j=1}^k \{v_{3j+1}\} \right)$$

as minimum secure dominating set. Upon removal of the remaining edges, in any order, the desired lower bound arises for $c_q(\mathcal{W}_{3k+1})$.

(c) Denote the hub of \mathcal{W}_{3k+2} by v_0 and the vertices of the remaining cycle \mathcal{C}_{3k+2} by v_1, \dots, v_{3k+2} . It follows that $X = \left(\bigcup_{j=1}^k \{v_{3j+1}\} \right) \cup \{v_0\}$ is a minimum secure dominating set of \mathcal{W}_{3k+2} by Theorem 6.9. The following four cases are considered in establishing the formula for $c_q(\mathcal{W}_{3k+2})$:

Case c(i): $0 \leq q \leq 2k + 1$. The set of vertices X remains a secure dominating set upon the removal of any q edges from the set $\{v_2v_3, v_5v_6, v_8v_9, \dots, v_{3k+2}v_1\} \cup \{v_0v_4, v_0v_7, v_0v_{10}, \dots, v_0v_{3k+1}\}$, showing that $c_q(\mathcal{W}_{3k+2}) = 0$ in this case.

Case c(ii): $2k + 1 < q \leq 2k + 3$. Removal of the $2k + 1$ edges as described in *Case c(i)* yields a graph comprising k four-cycles $\{v_0, v_{3j}, v_{3j+1}, v_{3j+2}\}$ for $j = 1, \dots, k$ and a triangle $\{v_0, v_1, v_2\}$ which are all joined to the vertex v_0 . This graph admits X as secure dominating set. Upon the removal of the edge v_0v_1 , the set $X \cup \{v_1\}$ admits minimum secure dominating set. This is followed by the removal of the edge v_0v_2 , in which case the set $X \cup \{v_1\}$ remains a secure dominating set.

Case c(iii): $2k + 3(\ell - 1) + 3 < q \leq 2k + 3\ell + 3$ for $\ell = 1, \dots, k$. Removal of the $2k + 3$ edges as described in *Cases c(i)* and *c(ii)* yields a graph comprising k four-cycles $\{v_0, v_{3j}, v_{3j+1}, v_{3j+2}\}$ for $j = 1, \dots, k$ joined by the vertex v_0 and a path of order 2 $\langle \{v_1, v_2\} \rangle$. This graph admits $X \cup \{v_1\}$ as minimum secure dominating set. By removing a further $3k$ edges in the order $v_0v_{3j}, v_0v_{3j+2}, v_{3j+1}v_{3j+2}$, starting with $j = 1, \dots, k$, the desired lower bound arises for $c_q(\mathcal{W}_{3k+2})$. In this case

$$X \cup \{v_1\} \cup \left(\bigcup_{j=1}^k \{v_{3j+2}\} \right)$$

is a minimum secure dominating set of the resulting.

Case c(iv): $5k + 3 + j \leq q \leq 6k + 4$ for $j = 1, \dots, k + 1$. Removal of the $5k + 3$ edges as described in *Cases c(i)–c(iii)* yields a graph comprising $(k + 1)\mathcal{P}_1 \cup (k + 1)\mathcal{P}_2$. In this case

$$X \cup \{v_1\} \cup \left(\bigcup_{j=1}^k \{v_{3j+2}\} \right)$$

admits a minimum secure dominating set. Upon removal of the remaining edges, in any order, the desired lower bound arises for $c_q(\mathcal{W}_{3k+2})$. ■

It is also possible to find good bounds on $C(\mathcal{W}_n)$.

Theorem 6.11 *Suppose $k \in \mathbb{N}$. Then,*

$$(a) \quad C_q(\mathcal{W}_{3k}) \geq \begin{cases} 0 & \text{if } 0 \leq q \leq 2 \\ \lfloor \frac{2q}{3} \rfloor - 1 & \text{if } 3 \leq q \leq 3k \\ 2k - 1 & \text{if } 3k < q \leq 6k - 1 \\ 2k & \text{if } q = 6k. \end{cases}$$

$$(b) \quad C_q(\mathcal{W}_{3k+1}) \geq \begin{cases} \lfloor \frac{2q}{3} \rfloor & \text{if } 0 \leq q \leq 3k+1 \\ 2k & \text{if } 3k+1 < q \leq 6k+1 \\ 2k+1 & \text{if } q = 6k+2. \end{cases}$$

$$(c) \quad C_q(\mathcal{W}_{3k+2}) \geq \begin{cases} \lfloor \frac{2q}{3} \rfloor & \text{if } 0 \leq q \leq 3k+2 \\ 2k+1 & \text{if } 3k+2 < q \leq 6k+3 \\ 2k+2 & \text{if } q = 6k+4. \end{cases}$$

Proof: Denote the hub of the wheel \mathcal{W}_{3k} by v_0 and the vertices of the remaining cycle \mathcal{C}_{3k} by v_1, \dots, v_{3k} . Note that $X = \left(\bigcup_{j=1}^k \{v_{3j-1}\}\right) \cup \{v_0\}$ is a minimum secure dominating set of \mathcal{W}_{3k} by Theorem 6.9. Consider the following four cases in establishing the formula for $C_q(\mathcal{W}_{3k})$:

Case a(i): $0 \leq q \leq 2$. By removing the edges v_1v_2 and $v_{3k}v_1$, the desired upper bound arises. In this case, a minimum secure dominating set is X .

Case a(ii): $3 \leq q \leq 3k$. By removing q edges in the order $v_2v_3, v_3v_4, \dots, v_{q+1}v_{q+2}, \dots, v_{3k-1}v_{3k}$, the desired upper bound arises, a minimum secure dominating set in this case being the union of X with the first $\lfloor 2q/3 \rfloor - 1$ vertices starting from v_2 to v_{3k} that are not in the set X .

Case a(iii): $3k < q \leq 6k - 1$. Removal of the $3k$ edges as described in *Cases a(i)* and *a(ii)* results in a graph that is isomorphic to the star $\mathcal{K}_{1,3k}$ for which $V(\mathcal{W}_{3k}) - \{v_1\}$ is a minimum secure dominating set. Upon removal of the remaining edges, in any order, the secure domination number remains unchanged and the desired lower bound arises for $C_q(\mathcal{W}_{3k})$.

Case a(iv): $q = 6k$. Removal of the $6k - 1$ edges described in *Case a(i)–a(iii)* yields a graph that is isomorphic to $(3k - 1)\mathcal{P}_1 \cup \mathcal{P}_2$ for which $V(\mathcal{W}_{3k}) - \{v_1\}$ is a minimum secure dominating set. Upon removal of the only remaining edge, the secure domination number increases by one and the desired lower bound arises.

(b) Denote the hub of \mathcal{W}_{3k+1} by v_0 and the vertices of the remaining cycle \mathcal{C}_{3k+1} by v_1, \dots, v_{3k+1} . It follows that $X = \left(\bigcup_{j=1}^k \{v_{3j}\}\right) \cup \{v_0\}$ is a minimum secure dominating set of \mathcal{W}_{3k+1} by Theorem 6.9. Consider the following three cases in establishing the formula for $C_q(\mathcal{W}_{3k+1})$:

Case b(i): $0 \leq q \leq 3k + 1$. By removing q edges in the order $v_1v_2, v_2v_3, v_3v_4, \dots, v_{q+1}v_{q+2}, \dots, v_{3k}v_{3k+1}, v_{3k+1}v_1$ the desired upper bound arises. In this case a minimum secure dominating set is the union of X with the first $\lfloor 2q/3 \rfloor$ vertices starting from v_2 to v_{3k+1} that are not in the set X .

Case b(ii): $3k + 1 < q \leq 6k + 1$. Removal of the $3k + 1$ edges as described in *Case b(i)*, results in a graph that is isomorphic to the star $\mathcal{K}_{1,3k+1}$ for which $V(\mathcal{W}_{3k+1}) - \{v_1\}$ is a minimum secure dominating set. Upon removal of the remaining edges, in any order, the secure domination number remains unchanged and the desired lower bound arises for $C_q(\mathcal{W}_{3k+1})$.

Case b(iii): $q = 6k + 2$. Removal of $6k + 1$ edges as described in *Cases b(i)* and *b(ii)* yields a graph that is isomorphic to $(3k)\mathcal{P}_1 \cup \mathcal{P}_2$ for which $V(\mathcal{W}_{3k+1}) - \{v_1\}$ is a minimum secure dominating set. Upon removal of the only remaining edge, the secure domination number increases by one and the desired lower bound arises.

(c) Denote the hub of \mathcal{W}_{3k+2} by v_0 and the vertices of the remaining cycle \mathcal{C}_{3k+2} by v_1, \dots, v_{3k+2} . It follows that $X = \left(\bigcup_{j=1}^k \{v_{3j+1}\}\right) \cup \{v_0\}$ is a minimum secure dominating set of \mathcal{W}_{3k+2} by Theorem 6.9. Consider the following three cases in establishing the formula for $C_q(\mathcal{W}_{3k+2})$:

Case c(i): $0 \leq q \leq 3k + 2$. By removing the q edges in the order $v_2v_3, v_3v_4, \dots, v_{q+1}v_{q+2}, \dots, v_{3k+1}v_{3k+2}, v_{3k+2}v_1, v_1v_2$, the desired upper bound arises. In this case, a minimum secure domi-

nating set is the union of X with the first $\lfloor 2q/3 \rfloor$ vertices starting from v_3 to v_{3k+2} (and finally v_2) that are not in the set X .

Case c(ii): $3k+2 < q \leq 6k+3$. Removal of the $3k+2$ edges as described in *Case c(i)*, results in a graph that is isomorphic to the star $\mathcal{K}_{1,3k+2}$ for which $V(\mathcal{W}_{3k+2}) - \{v_1\}$ is a minimum secure dominating set. Upon removal of the remaining edges, in any order, the secure domination number remains unchanged and the desired lower bound arises for $C_q(\mathcal{W}_{3k+2})$.

Case c(iii): $q = 6k+4$. Removal of the $6k+3$ edges described in *Cases c(i)* and *c(ii)* yields a graph that is isomorphic to $(3k+1)\mathcal{P}_1 \cup \mathcal{P}_2$ for which $V(\mathcal{W}_{3k+2}) - \{v_1\}$ is a minimum secure dominating set. Upon removal of the only remaining edge, the secure domination number increases by one and the desired lower bound arises. ■

6.5 The cost functions c_q and C_q for complete bipartite graphs

It follows from Corollaries 6.2 and 6.4 that $n - (j+1)(n-j) + q \leq c_q(\mathcal{K}_{j,n-j}) \leq C_q(\mathcal{K}_{j,n-j}) \leq q$ for all $n-j \geq j$ and $0 \leq q \leq n$, by noting that $\chi(\overline{\mathcal{K}_{j,n-j}}) = \chi(\mathcal{K}_j \cup \mathcal{K}_{n-j}) = n-j$. Again, these bounds seem to be weak for small values of q .

For the simplest class of complete bipartite graphs, namely stars, it is possible to determine the values of c and C exactly. The proof of the following straight-forward result is omitted.

Theorem 6.12 (The sequences c and C for stars) *For the star $\mathcal{K}_{1,n-1}$ of order n ,*

$$c_q(\mathcal{K}_{1,n-1}) = C_q(\mathcal{K}_{1,n-1}) = \begin{cases} 0 & \text{if } 0 \leq q \leq n-2 \\ 1 & \text{if } q = n-1. \end{cases}$$

In the next result perhaps the simplest and most natural bipartite generalisation of a star, namely the graph $\mathcal{K}_{2,n-2}$, is considered.

Theorem 6.13 (The sequences c and C for the complete bipartite graph $\mathcal{K}_{2,n-2}$)

For the complete bipartite graph $\mathcal{K}_{2,n-2}$ of order $n \geq 4$, $c_q(\mathcal{K}_{2,n-2}) = \lfloor q/2 \rfloor$ for all $0 \leq q \leq 2n-4$ and

$$C_q(\mathcal{K}_{2,n-2}) = \begin{cases} 0 & \text{if } q = 0 \\ q-1 & \text{if } 1 \leq q \leq n-2 \\ n-3 & \text{if } n-1 \leq q \leq 2n-5 \\ n-2 & \text{if } q = 2n-4. \end{cases}$$

Proof: Denote the partite sets of $\mathcal{K}_{2,n-2}$ by $\{x, y\}$ and $V = \{v_1, \dots, v_{n-2}\}$. Removing q edges from $\mathcal{K}_{2,n-2}$ results in a subgraph $G \in \mathcal{K}_{2,n-2} - qe =: K(n, q)$ and the partition $V = V_0^G \cup V_x^G \cup V_y^G \cup V_{xy}^G$, where V_0^G contains isolated vertices in G , V_x^G (V_y^G , resp.) contains the vertices adjacent to x only (y only, resp.) in G , and V_{xy}^G contains the common neighbours of x and y in G . Then, $2|V_0^G| + |V_x^G| + |V_y^G| = q$, so that

$$|V_0^G| + |V_x^G| + |V_y^G| = q - |V_0^G|. \quad (6.2)$$

In order to determine a minimum secure dominating set for G , three mutually exclusive cases are considered, using the notation \tilde{A} to denote all but one elements of a set A of indistinguishable elements.

Case i: Both V_x^G and V_y^G are empty. In this case G is the vertex disjoint union of the isolated vertices in V_0^G and either a complete bipartite graph with $\{x, y\}$ as one of its partite sets or the

two further isolated vertices x and y , as shown in Figure 6.3. In both of these subcases G is securely dominated by the vertices in $V_0^G \cup \{x, y\}$, and no smaller secure dominating set of G exists by Proposition 3.7.

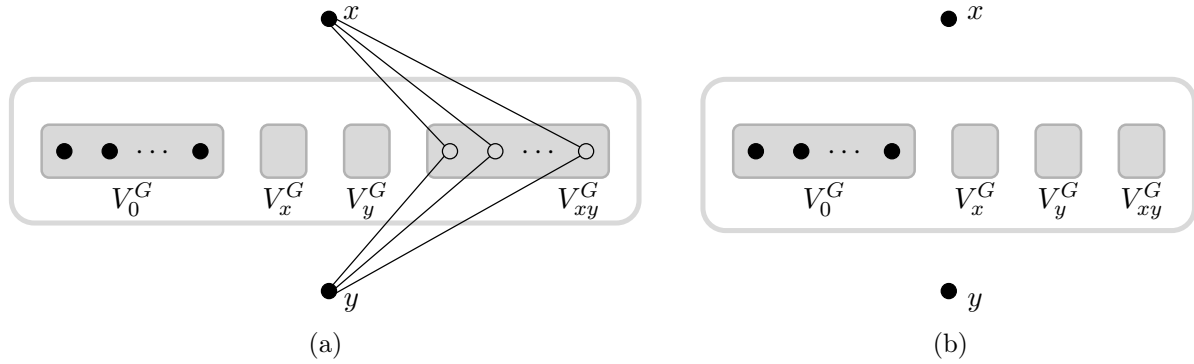


FIGURE 6.3: The secure dominating set X^G of minimum cardinality $\gamma_s(G)$, denoted by solid vertices, for case i, where $V_x^G = V_y^G = \emptyset$. (a) The situation where $V_{xy}^G \neq \emptyset$. (b) The situation where $V_{xy}^G = \emptyset$.

Case ii: Neither V_x^G nor V_y^G is empty, but V_{xy}^G is empty, as shown in Figure 6.4. In this case G is the vertex disjoint union of the isolated vertices in V_0^G and two vertex disjoint stars centred at x and y , respectively. Therefore G is securely dominated by the vertices in $V_0^G \cup V_x^G \cup V_y^G$, and no smaller secure dominating set of G exists by Proposition 3.7.

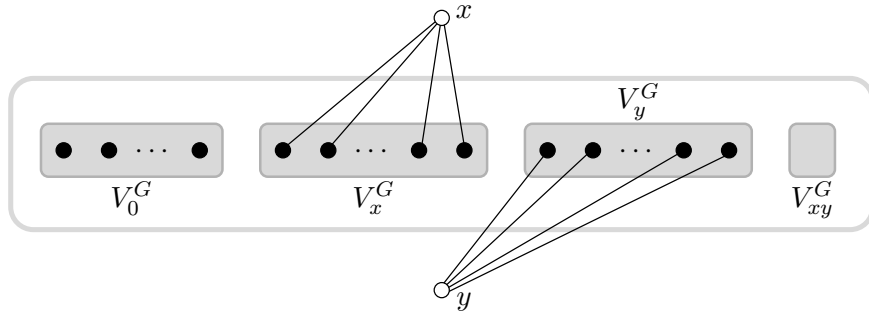


FIGURE 6.4: A secure dominating set X^G of minimum cardinality $\gamma_s(G)$, denoted by solid vertices, for case ii where $V_x^G \neq \emptyset$ and $V_y^G \neq \emptyset$.

Case iii: (a) At least one of V_x^G or V_y^G is empty, or (b) none of V_x^G , V_y^G or V_{xy}^G is empty, as shown in Figure 6.5. In both these subcases, at least two vertices in the set $V_{xy}^G \cup \{x, y\}$ are required to securely dominate the subgraph of G induced by $V_{xy}^G \cup \{x, y\}$ due to Proposition 3.7. Although the stars induced by $\{x\} \cup V_x^G$ and $\{y\} \cup V_y^G$ are dominated securely by the minimum secure dominating sets

$$S_x = \begin{cases} \{x\} \cup \widetilde{V_x^G}, & \text{if } V_x^G \neq \emptyset \\ \{x\}, & \text{if } V_x^G = \emptyset \end{cases} \quad \text{and} \quad S_y = \begin{cases} \{y\} \cup \widetilde{V_y^G}, & \text{if } V_y^G \neq \emptyset \\ \{y\}, & \text{if } V_y^G = \emptyset, \end{cases}$$

respectively, according to Proposition 3.7, one of the full sets V_x^G or V_y^G is nevertheless required in subcase (b) in any secure dominating set of G in order to allow for a swap set involving one of the vertices in $\{x, y\}$ and one of the vertices in V_{xy}^G .

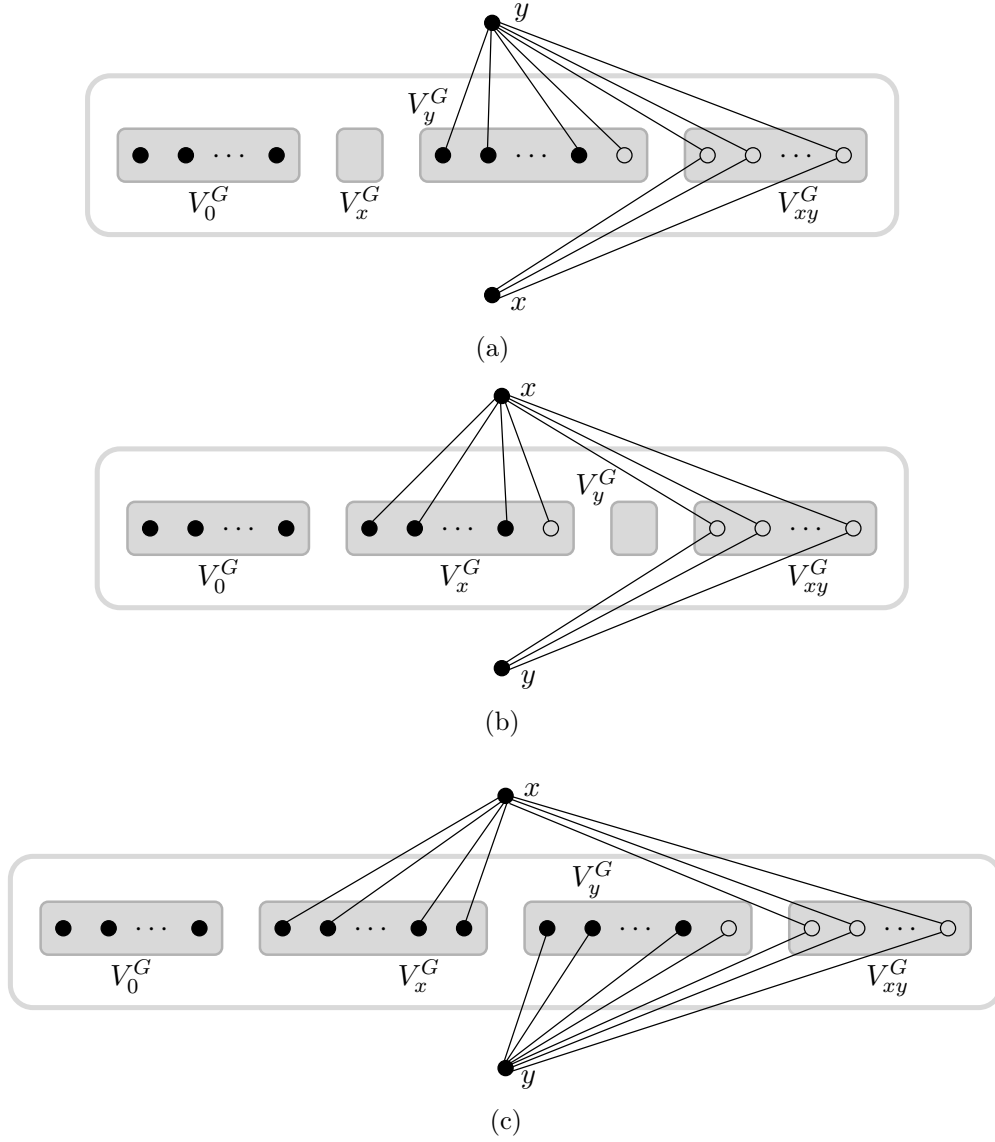


FIGURE 6.5: A secure dominating set X^G of minimum cardinality $\gamma_s(G)$, denoted by solid vertices, for the third case. (a) and (b) The situation in case iii(a) where at least one of V_x^G or V_y^G is empty. (c) The situation where the full set V_x^G is required to dominate the graph securely.

From the above three cases it follows that

$$X^G = \begin{cases} V_0^G \cup \{x, y\}, & \text{if } V_x^G = \emptyset, V_y^G = \emptyset \\ V_0^G \cup V_x^G \cup V_y^G, & \text{if } V_x^G \neq \emptyset, V_y^G \neq \emptyset, V_{xy}^G = \emptyset \\ V_0^G \cup \widetilde{V_x^G} \cup V_y^G, & \text{otherwise} \end{cases}$$

is a minimum secure dominating set for G and hence that

$$\gamma_s(G) = \begin{cases} q - |V_0^G| + 2, & \text{if } V_x^G = \emptyset, V_y^G = \emptyset \\ q - |V_0^G|, & \text{if } V_x^G \neq \emptyset, V_y^G \neq \emptyset, V_{xy}^G = \emptyset \\ q - |V_0^G| + 1, & \text{otherwise} \end{cases} \quad (6.3)$$

by (6.2). Since q is fixed, it follows by (6.3) that $\gamma_s(G)$ is minimised (maximised, resp.) by maximising (minimising, resp.) the quantity $|V_0^G|$.

The number of vertices in V_0^G is maximised by removing from $\mathcal{K}_{2,n-2}$ the edges $xv_1, yv_1, xv_2, yv_2, xv_3, yv_3$, and so on, in this order, until q edges have been removed. In this way, $|V_0^G| = (q-1)/2$, $|V_x^0| = 0$, $|V_y^0| = 1$ and $|V_{xy}^G| = n - (q+5)/2$ if q is odd, while $|V_0^G| = q/2$, $|V_x^0| = |V_y^0| = 0$ and $|V_{xy}^G| = n - (q+4)/2$ if q is even. It follows by the first and third cases of (6.3) that

$$\begin{aligned} c_q(\mathcal{K}_{2,n-2}) &= \min_{G \in K(n,q)} \{\gamma_s(G)\} - 2 \\ &= \begin{cases} q - \frac{q-1}{2} - 1, & \text{if } q \text{ is odd} \\ q - \frac{q}{2}, & \text{if } q \text{ is even} \end{cases} \\ &= \lfloor q/2 \rfloor. \end{aligned}$$

Clearly, $C_0(\mathcal{K}_{2,n-2}) = 0$. If $0 < q \leq n-2$, then the number of vertices in V_0^G is minimised by removing from $\mathcal{K}_{2,n-2}$ the edges xv_1, xv_2, xv_3 , and so on, in this order, until q edges have been removed. In this way, $|V_0^G| = |V_x^0| = 0$, $|V_y^0| = q$ and $|V_{xy}^G| = n - q - 2$, resulting in the expression

$$C_q(\mathcal{K}_{2,n-2}) = \max_{G \in K(n,q)} \{\gamma_s(G)\} - 2 = q - 1, \quad \text{if } 1 \leq q \leq n - 2$$

when substituted into the third case of (6.3). If $n-2 < q \leq 2n-5$, then the number of vertices in V_0^G is minimised by removing the edges $xv_1, xv_2, \dots, xv_{n-2}$ together with the edges yv_1, yv_2, yv_3 , and so on, in this order, until q edges have been removed. In this way, $|V_0^G| = q - (n-2)$, $|V_x^G| = 0$, $|V_y^G| = (2n-4) - q$ and $|V_{xy}^G| = 0$, resulting in the expression

$$C_q(\mathcal{K}_{2,n-2}) = \max_{G \in K(n,q)} \{\gamma_s(G)\} - 2 = n - 3, \quad \text{if } n-1 \leq q \leq 2n-5$$

when substituted into the third case of (6.3). Finally, $K(n, 2n-4)$ contains only the empty graph of order n , so that $C_{2n-4}(\mathcal{K}_{2,n-2}) = n-2$. ■

It seems rather difficult to generalise the above result for the graph $\mathcal{K}_{j,n-j}$, where $j > 2$, because of the large number of cases involved in a generalisation of the proof of Theorem 6.13. However, the following good upper bounds on the sequences $\mathbf{c}(\mathcal{K}_{j,n-j})$ and $\mathbf{C}(\mathcal{K}_{j,n-j})$ are established for $j = 3, 4$.

Theorem 6.14 *If $j \in \{2, 3, 4\}$, then*

$$c_q(\mathcal{K}_{j,n-j}) \leq \begin{cases} 0 & \text{if } 0 \leq q \leq (j-2)(n-j) \\ \left\lfloor \frac{q-(j-2)(n-j)}{2} \right\rfloor & \text{if } (j-2)(n-j) < q \leq j(n-j) \end{cases}$$

and

$$C_q(\mathcal{K}_{j,n-j}) \geq \begin{cases} 0 & \text{if } 0 \leq q \leq (j-2)(n-j) \\ q - (j-2)(n-j) - 1 & \text{if } (j-2)(n-j) < q \leq (j-1)(n-j) \\ n - j - 1 & \text{if } (j-1)(n-j) < q \leq j(n-j) - 1 \\ n - j & \text{if } q = j(n-j). \end{cases}$$

Proof: Equality holds for $j = 2$ in all of the above formulae by Theorem 6.13. Therefore, suppose $j \in \{3, 4\}$ and denote the partite sets of $\mathcal{K}_{j,n-j}$ by $X = \{x_1, \dots, x_j\}$ and $V = \{v_1, \dots, v_{n-j}\}$. Note that X is a minimum secure dominating set of $\mathcal{K}_{j,n-j}$ by Proposition 3.7.

If $j = 3$ and $0 \leq q \leq n-3$, then removal of the edges $x_1v_1, x_1v_2, \dots, x_1v_q$ from $\mathcal{K}_{3,n-3}$ results in a subgraph of $\mathcal{K}_{3,n-3}$ which, in turn, contains $\mathcal{K}_{2,n-3} \cup \langle \{x_1\} \rangle$ as subgraph, for which X is

still a minimum secure dominating set. If $j = 4$ and $0 \leq q \leq 2(n - 4)$, removal of the edges $x_1v_1, x_1v_2, \dots, x_1v_{q'}$ and $x_2v_1, x_2v_2, \dots, x_2v_{q''}$ such that $q = q' + q''$ results in a subgraph of $\mathcal{K}_{4,n-4}$ which, in turn, contains $\mathcal{K}_{2,n-4} \cup \{x_1, x_2\}$, for which X is again a minimum secure dominating set. This shows that $c_q(\mathcal{K}_{j,n-j}) = 0$ for all $0 \leq q \leq (j - 2)(n - j)$, where $j \in \{3, 4\}$.

If $j = 3$ and $n - 3 < q \leq 3(n - 3)$, then removal of the edges $x_1v_1, x_1v_2, \dots, x_1v_{n-j}$ results in a subgraph of $\mathcal{K}_{3,n-3}$ containing $\mathcal{K}_{2,n-3} \cup \{x_1\}$, for which X is still a minimum secure dominating set. A further $q - n + 3$ edges may be removed, as described in the proof of Theorem 6.13, where $\{x_2, x_3\}$ fulfils the role of $\{x, y\}$. If $j = 4$ and $2(n - 4) < q \leq 4(n - 4)$, then removal of the edges $x_1v_1, x_1v_2, \dots, x_1v_{n-4}$ and $x_2v_1, x_2v_2, \dots, x_2v_{n-4}$ results in the subgraph of $\mathcal{K}_{4,n-4}$ containing $\mathcal{K}_{2,n-4} \cup \{x_1, x_2\}$, for which X is still a minimum secure dominating set. A further $q - 2(n - 4)$ edges may be removed as described in the proof of Theorem 6.13, where $\{x_3, x_4\}$ fulfils the roles of $\{x, y\}$. This results in the desired upper bound for $c_q(\mathcal{K}_{j,n-j})$.

For $0 \leq q \leq (j - 2)(n - j)$ with $j = \{3, 4\}$, the removal of edges from $C_q(\mathcal{K}_{j,n-j})$ occur in the same manner as in the proof of the bound on $c_q(\mathcal{K}_{j,n-j})$, resulting in $C_q(\mathcal{K}_{j,n-j}) = c_q(\mathcal{K}_{j,n-j}) = 0$, for which X is still a minimum secure dominating set.

If $j = 3$ and $n - 3 < q \leq 3(n - 3)$, then removal of the edges $x_1v_1, x_1v_2, \dots, x_1v_{n-j}$ results in a subgraph of $\mathcal{K}_{3,n-3}$ containing $\mathcal{K}_{2,n-3} \cup \{x_1\}$, for which X is still a minimum secure dominating set. A further $q - n + 3$ edges may be removed, as described in the proof of Theorem 6.13, where $\{x_2, x_3\}$ fulfils the role of $\{x, y\}$. If $j = 4$ and $2(n - 4) < q \leq 4(n - 4)$, then removal of the edges $x_1v_1, x_1v_2, \dots, x_1v_{n-4}$ and $x_2v_1, x_2v_2, \dots, x_2v_{n-4}$ results in the subgraph of $\mathcal{K}_{4,n-4}$ containing $\mathcal{K}_{2,n-4} \cup \{x_1, x_2\}$, for which X is still a minimum secure dominating set. A further $q - 2(n - 4)$ edges may be removed, as described in the proof of Theorem 6.13, where $\{x_3, x_4\}$ fulfils the roles of the vertices $\{x, y\}$, which results in the desired upper bound for $C_q(\mathcal{K}_{j,n-j})$. ■

Although the bounding sequences in Theorem 6.14 are expected to be good approximations of the sequences for $\mathbf{c}(\mathcal{K}_{j,n-j})$ and $\mathbf{C}(\mathcal{K}_{j,n-j})$ when $j = 3, 4$, these approximations are not exact. For example, the upper bound $c_6(\mathcal{K}_{3,3}) \leq 1$ follows from Theorem 6.14 while, in fact, $c_6(\mathcal{K}_{3,3}) = 0$ as may be seen in Figure 6.6(a). Similarly, Theorem 6.14 yields the lower bound $C_3(\mathcal{K}_{3,4}) \geq 0$, while, in fact, $C_3(\mathcal{K}_{3,4}) \geq 1$ as may be seen in Figure 6.6(b).



FIGURE 6.6: Examples showing that the bounding sequences resulting from Theorem 6.14 are not exact. It is shown in (a) that $c_6(\mathcal{K}_{3,3}) = 0$ and in (b) that $C_3(\mathcal{K}_{3,4}) \geq 1$, where the solid vertices denote members of a minimum secure dominating set in each case.

It is also possible to establish the following bounds on the sequences $\mathbf{c}(\mathcal{K}_{j,n-j})$ and $\mathbf{C}(\mathcal{K}_{j,n-j})$ for $j \geq 5$.

Theorem 6.15 For the complete bipartite graph $\mathcal{K}_{j,n-j}$ with $j \geq 5$,

$$c_q(\mathcal{K}_{j,n-j}) \leq \begin{cases} 0 & \text{if } 0 \leq q \leq 8 + (j - 2)n - j^2 \\ \left\lfloor \frac{q + j^2 - (j - 2)n - 8}{2} \right\rfloor & \text{if } 8 + (j - 2)n - j^2 < q \leq j(n - j) \end{cases}$$

and

$$C_q(\mathcal{K}_{j,n-j}) \geq \begin{cases} \ell & \text{if } \ell(n-j) \leq q \leq (\ell+1)(n-j)-1, \\ & \text{for } \ell = 0, 1, \dots, j-5 \\ j-4 & \text{if } (j-4)(n-j) \leq q \leq (j-2)(n-j) \\ q - (j-2)(n-j+1) - 7 & \text{if } (j-2)(n-j) < q \leq (j-1)(n-j) \\ n-j-1 & \text{if } (j-1)(n-j) < q \leq j(n-j)-1 \\ n-j & \text{if } q = j(n-j). \end{cases}$$

Proof: Denote the partite sets of $\mathcal{K}_{j,n-j}$ by $X = \{x_1, \dots, x_j\}$ and $V = \{v_1, v_2, \dots, v_{n-j}\}$, and note that $\{x_1, x_2, v_1, v_2\}$ is a minimum secure dominating set for $\mathcal{K}_{j,n-j}$ by Proposition 3.7. Consider the following two cases in establishing the formula for $c_j(\mathcal{K}_{j,n-j})$:

Case i: $0 \leq q \leq 8 + (j-2)n + j^2$. The set of vetices $\{x_1, x_2, v_1, v_2\}$ remains a minimum secure dominating set upon the removal of any q of the edges from the set $\{x_1v_1, x_1v_2, x_2v_1, x_2v_2\} \cup \{x_iv_k \mid i = 3, 4, \dots, j \text{ and } k = 3, 4, \dots, n-j\}$, showing that $c_q(\mathcal{K}_{j,n-j}) = 0$ in this case.

Case ii: $8 + (j-2)n + j^2 < q \leq j(n-j)$. If all $(j-2)(n-j-2) + 4 = 8 + (j-2)n + j^2$ edges described in *Case i* above are removed, then the resulting graph comprises two disjoint subgraphs isomorphic to $\mathcal{K}_{2,j-2}$ and $\mathcal{K}_{2,n-j-2}$, respectively. By removing a further $q - (8 + (j-2)n + j^2)$ edges from these subgraphs, as described in the proof of Theorem 6.13, first from one of the subgraphs and then from the other, the desired upper bound arises for $c_q(\mathcal{K}_{j,n-j})$.

Finally, consider the following three cases in establishing the lower bound on $C_j(\mathcal{K}_{j,n-j})$:

Case a: $\ell(n-j) \leq q \leq (\ell+1)(n-j)-1$ for $\ell = 0, 1, \dots, j-5$. Removing $\ell(n-j)$ edges from the set $\{x_iv_k \mid i = 5, 6, \dots, 5+\ell \text{ and } k = 1, 2, \dots, n-j\}$ and a further $q - \ell(n-j)$ edges from the set $\{x_iv_k \mid i = 6+\ell, 7+\ell, \dots, j \text{ and } k = 3, \dots, n-j\}$ results in the set $\{x_1v_1, x_1v_2, x_2v_1, x_2v_2\} \cup \{x_5, x_6, \dots, x_{5+\ell}\}$ being a minimum secure dominating set for the resulting graph, showing that $C_q(\mathcal{K}_{j,n-j}) \geq \ell$ in this case.

Case b: $(j-4)(n-j) \leq q \leq (j-2)(n-j)$. The removal of $(j-4)(n-j)$ edges from the set $\{x_iv_k \mid i = 5, 6, \dots, j \text{ and } k = 1, 2, \dots, n-j\}$ yields the subgraph $\mathcal{K}_{4,n-j} \cup \langle \{x_5, \dots, x_j\} \rangle$. The remaining $q - (j-4)(n-j)$ edges may be removed from the set $\{x_iv_k \mid i = 3, 4 \text{ and } k = 1, 2, \dots, n-j\}$ resulting in the set X being a minimum secure dominating set for the resulting graph. This shows that $C_q(\mathcal{K}_{j,n-j}) \geq j-4$ for this case.

Case c: $(j-2)(n-j) < q \leq j(n-j)$. The removal of $(j-2)(n-j)$ edges from the set $\{x_iv_k \mid i = 3, 4, \dots, j \text{ and } k = 1, 2, \dots, n-j\}$ yields a graph which contains $\mathcal{K}_{2,n-j} \cup \langle \{x_3, x_4, \dots, x_j\} \rangle$ as subgraph for which X is a minimum secure dominating set. By removing a further $q - (j-2)(n-j)$ edges from the subgraph, as described in the proof of Theorem 6.13, the desired upper bound results for $C_q(\mathcal{K}_{j,n-j})$. ■

The bounding sequences in Theorem 6.15 are again expected to be good approximations of the sequences for $c(\mathcal{K}_{j,n-j})$ and $C(\mathcal{K}_{j,n-j})$ for $j \geq 5$, but these approximations are known not to be exact.

6.6 The cost functions c_q and C_q for complete graphs

It follows by Corollaries 6.2 and 6.4 that $n - \binom{n}{2} + q - 1 \leq c_q(\mathcal{K}_n) \leq C_q(\mathcal{K}_n) \leq q$, but these bounds are weak for small q . A greedy approach towards establishing a good upper bounding sequence

on $c(\mathcal{K}_n)$ seems to be to remove edges from \mathcal{K}_n successively in such a manner that an increase in the cardinality of a minimum secure dominating set for the resulting graph is delayed as long as possible. This section opens by putting forward such a recursive edge removal strategy. Note that $c_q(\mathcal{K}_n) \geq 1$ for all $n \geq 2$ and all $0 < q \leq \binom{n}{2}$. The successive edge removal strategy attempts to maintain an edge critical cost value of 1 as long as possible, before decomposing the problem of approximating the value of $c_q(\mathcal{K}_n)$ into the two smaller subproblems of approximating the values of c_q for two disjoint cliques within \mathcal{K}_n which may each, in turn, be tackled by iteratively applying the same successive edge removal procedure, and so on.

Suppose n is even, so that $n = 2r$ for some $r \in \mathbb{N}$. Let $v_0, v_1, \dots, v_{2r-1}$ be the vertices of \mathcal{K}_n and let \mathcal{F}_ℓ be the 1-factor of \mathcal{K}_n containing the r edges $v_j v_k$ for which $k + j \equiv \ell \pmod{n}$, for all $\ell = 0, 1, \dots, r-1$. Then the edges contained in the 1-factors $\mathcal{F}_0, \mathcal{F}_1, \dots, \mathcal{F}_{r-1}$ are exactly those edges of \mathcal{K}_n between pairs of vertices with indices of different parity. Hence the graph

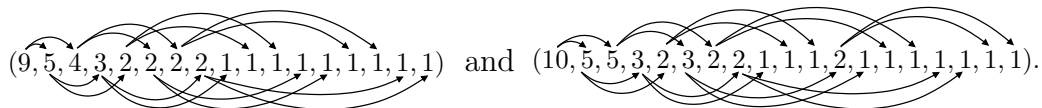
$$\mathcal{K}_{2r} - \bigoplus_{\ell=0}^{r-1} \mathcal{F}_\ell, \quad (6.4)$$

obtained by removing from \mathcal{K}_n all the edges contained in these 1-factors, comprises two vertex disjoint cliques of order r , one induced by the vertices with even indices and the other induced by the vertices with odd indices.

If n is odd, then $n = 2r - 1$ for some $r \in \mathbb{N}$, in which case a dummy vertex may be added to \mathcal{K}_n , joined to each of the vertices of \mathcal{K}_n , in order to form a graph isomorphic to \mathcal{K}_{2r} . Then the above successive edge removal procedure may be applied to this enlarged graph, after which the dummy vertex (and all edges incident to it) may be removed from one of the two resulting vertex disjoint subgraphs in (6.4), giving rise to two vertex disjoint cliques of \mathcal{K}_n , one of order r and the other of order $r - 1$.

Regardless of the parity of n , the above edge removal procedure will result in two cliques, of orders $\lceil \frac{n}{2} \rceil$ and $\lfloor \frac{n}{2} \rfloor$, after $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ edge removals from \mathcal{K}_n . Each of these cliques may be defended by a single vertex in the clique, resulting in an increase of only one in the cardinality of a minimum secure dominating set of the resulting graph after the removal of these $\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor$ edges. At this point the problem is decomposed into two smaller subproblems, as described above, and each time a new subproblem is encountered the upper bound on the cardinality of a minimum secure dominating set of the resulting graph increases by one. This process is repeated until a subproblem is reached in which a clique of order 1 is considered. A pseudo-code listing of this iterative procedure is given in the guise of a breadth-first search as Algorithm 6.1.

Algorithm 6.1 maintains two lists during execution. The first list, `Traversallist`, contains the orders of the shrinking disjoint cliques into which the original problem is decomposed, as explained above. For example, for $n = 9$ and $n = 10$, these lists are respectively



The second list is called `cBoundSequence`. This list is populated with appropriate upper bounds on $c_q(\mathcal{K}_n)$ during execution of the algorithm. For example, `cBoundSequence` is

$(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 5, 6, 7, 8)$ and
 $(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 5, 6, 6, 7, 8, 9)$

for $n = 9$ and $n = 10$, respectively. The following lower bounding sequence on $C(\mathcal{K}_n)$ may also be established.

Algorithm 6.1: An upper bound on the sequence $c(\mathcal{K}_n)$ for a complete graph of order n

Input : The graph order, n .
Output: An upper bound sequence cBoundSequence on $c(\mathcal{K}_n)$.

```

1 TraversalList  $\leftarrow (n)$ , TraversalPosition  $\leftarrow 1$ ;
2 cBoundSequence  $\leftarrow (0)$ , cValue  $\leftarrow 0$ ;
3 while TraversalPosition  $\leq |\text{TraversalList}|$  do
4    $x \leftarrow \text{TraversalList}[\text{TraversalPosition}]$ ;
5   cValue  $\leftarrow \text{cValue} + 1$ ;
6   if  $x > 1$  then
7     Append(cBoundSequence,  $\binom{x}{2} - \binom{\lceil \frac{x}{2} \rceil}{2} - \binom{\lfloor \frac{x}{2} \rfloor}{2}$  copies of cValue);
8     Append(TraversalList,  $\lceil \frac{x}{2} \rceil$ );
9     Append(TraversalList,  $\lfloor \frac{x}{2} \rfloor$ );
10  TraversalPosition  $\leftarrow \text{TraversalPosition} + 1$ ;
    
```

Theorem 6.16 (A lower bound on the sequence C for a complete graph)

Suppose $n \in \mathbb{N}$. Let q be a nonnegative integer not exceeding $\binom{n}{2}$. Then

$$C_q(\mathcal{K}_n) \geq n - \sqrt{2\binom{n}{2} + 1 - 2q}. \quad (6.5)$$

Proof: The vertices of a complete graph may be isolated sequentially by successive edge removals in a greedy bid to produce a good lower bounding sequence on $C(\mathcal{K}_n)$. Let $\{v_1, v_2, \dots, v_n\}$ be the vertex set of \mathcal{K}_n . Then v_1 may be isolated by $n - 1$ edge removals, v_2 may be isolated by a further $n - 2$ edge removals, and so on, until the empty graph $\overline{\mathcal{K}_n}$ is obtained. Each time a vertex is thus isolated, the cardinality of a minimum secure dominating set of the resulting graph increases by one, resulting in the following lower bounding sequence on $C(\mathcal{K}_n)$: a single zero, followed by $n - 1$ ones, followed by $n - 2$ twos, followed by $n - 3$ threes, and so on, until the sequence terminates in a single occurrence of the value $n - 1$. This sequence is approximated by the formula on the righthand side of (6.5) to within an error of strictly less than $\frac{1}{2}$. ■

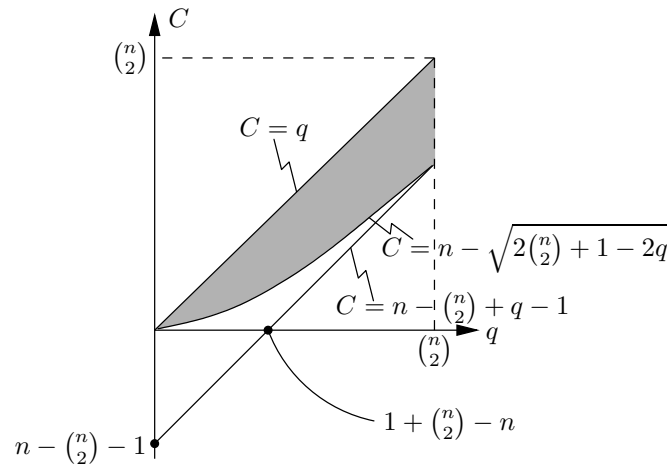


FIGURE 6.7: A graphical illustration of the bound on $C_q(\mathcal{K}_n)$ in Theorem 6.16. It follows from the theorem that the graph of C_q is a step-function of q which lies entirely within the shaded region.

A graphical representation of the bound in the theorem above may be found in Figure 6.7. Algorithm 6.1 and Theorem 6.16 yield, for example, the bounding sequences

$$\begin{aligned} \mathbf{c}(\mathcal{K}_6) &\leq (0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 3, 4, 5), \text{ and} \\ \mathbf{C}(\mathcal{K}_6) &\geq (0, 1, 1, 1, 1, 1, 1, 2, 2, \mathbf{2}, 3, 3, 3, 3, 4, 5). \end{aligned}$$

and

$$\begin{aligned} \mathbf{c}(\mathcal{K}_8) &\leq (0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \underline{2}, 2, 2, 2, 3, 3, 3, 3, 4, 5, 6, 7), \text{ and} \\ \mathbf{C}(\mathcal{K}_8) &\geq (0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 7). \end{aligned}$$

Although the bounding sequences in Algorithm 6.1 and Theorem 6.16 are expected to be good approximations of the sequences $\mathbf{c}(\mathcal{K}_n)$ and $\mathbf{C}(\mathcal{K}_n)$, these approximations are not exact. For example, the upper bounding sequence on $\mathbf{C}(\mathcal{K}_6)$ yields the upper bound $C_9(\mathcal{K}_6) \geq 2$ (shown in bold face above), while, in fact, $C_9(\mathcal{K}_6) \geq 3$ as may be seen in Figure 6.8(a). Similarly, the lower bounding sequence on $\mathbf{c}(\mathcal{K}_8)$ yields the upper bound $c_{17}(\mathcal{K}_8) \leq 2$ (underlined above) while, in fact, $c_{17}(\mathcal{K}_8) = 1$ as may be seen in Figure 6.8(b).

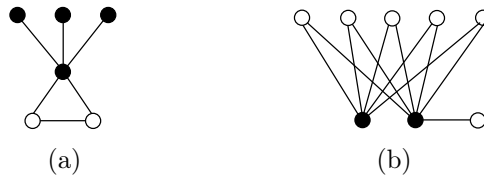


FIGURE 6.8: Examples showing that the bounding sequences resulting from Algorithm 6.1 and Theorem 6.16 are not exact. It is shown in (a) that $C_9(\mathcal{K}_6) \geq 3$ and in (b) that $c_{17}(\mathcal{K}_8) = 1$, where the solid vertices denote members of a minimum secure dominating set in each case.

6.7 Chapter summary

The effects of multiple edge failures on the secure domination number of a graph was explored in this chapter. The chapter opened in §6.1 with a brief reference to practical applications of secure domination, as well as the effect that edge failures may have on the potential effectiveness with which a graph can be dominated securely. Two cost functions, $c_q(G)$ and $C_q(G)$, were introduced for measuring respectively the smallest possible and largest possible increase of a minimum secure dominating set of a member of the set $G - qe$ over and above the value of $\gamma_s(G)$. The growth properties of the cost sequences $\mathbf{c}(G)$ and $\mathbf{C}(G)$ were analysed in Theorem 6.2.

Some general bounds on the secure domination number were presented in §6.2, and these bounds were used to derive lower bounds on c_q and upper bounds on C_q . The inequality chain $n - m + q + \chi(\overline{G}) \leq c_q(G) \leq C_q(G) \leq q$ was also established for any graph G of order n and size m .

The cost sequences $\mathbf{c}(\mathcal{P}_n)$, $\mathbf{C}(\mathcal{P}_n)$, $\mathbf{c}(\mathcal{C}_n)$ and $\mathbf{C}(\mathcal{C}_n)$ were determined exactly for paths and cycles in §6.3. The sequences for paths were established using a series of preliminary results showing in which manner a single edge may be removed from \mathcal{P}_n to ensure that $\gamma_s(\mathcal{P}_n - qe)$ is either maximised or minimised. The sequences $\mathbf{c}(\mathcal{P}_n)$ and $\mathbf{C}(\mathcal{P}_n)$ were then established by induction over q . The values of $\mathbf{c}(\mathcal{C}_n)$ and $\mathbf{C}(\mathcal{C}_n)$ follow trivially since the set $\mathcal{C}_n - 1e$ contains only a path of order n .

The secure domination number of a wheel \mathcal{W}_n of order n was determined in §6.4 using a result of Cockayne [32, Theorem 4]. Upper and lower bounds on $c_q(\mathcal{W}_n)$ and $C_q(\mathcal{W}_n)$, respectively, were also presented.

Complete bipartite graphs were considered in §6.5. The exact values of $\mathbf{c}(\mathcal{K}_{1,n-1})$ and $\mathbf{C}(\mathcal{K}_{1,n-1})$ were noted, after which the values of $\mathbf{c}(\mathcal{K}_{2,n-2})$ and $\mathbf{C}(\mathcal{K}_{2,n-2})$ were established. If the partite sets of $\mathcal{K}_{2,n-2}$ are denoted by $X = \{x, y\}$ and $V = \{v_1, \dots, v_{n-2}\}$, the set X is a minimum secure dominating set of $\mathcal{K}_{2,n-2}$. Removing the edges $xv_1, yv_1, xv_2, yv_2, xv_3, yv_3$, and so on establishes the values of $c_q(\mathcal{K}_{2,n-2})$ for $0 \leq q \leq 2n - 4$. The sequence for $\mathbf{C}(\mathcal{K}_{2,n-2})$ is determined by iteratively isolating x , followed by the isolation of y . Good upper and lower bounds were also provided for $\mathbf{c}(\mathcal{K}_{j,n-j})$ and $\mathbf{C}(\mathcal{K}_{j,n-j})$, respectively, for some $j > 2$.

The focus in §6.6 shifted towards establishing bounds on the cost sequences $\mathbf{c}(\mathcal{K}_n)$ and $\mathbf{C}(\mathcal{K}_n)$ for the complete graph of order n by repeatedly removing 1-factors from \mathcal{K}_n , resulting in two cliques $\mathcal{K}_{\lceil n/2 \rceil}$ and $\mathcal{K}_{\lfloor n/2 \rfloor}$, thereby decomposing the computation problem into two smaller subproblems. Each time a new subproblem is encountered the edge associated cost value increases by one. A lower bound was also established on $C_q(\mathcal{K}_n)$, by iteratively isolating vertices, until $\overline{\mathcal{K}_n}$ is obtained.

CHAPTER 7

Criticality and stability in secure graph domination

Contents

7.1	The notions of p -stability and q -criticality	137
7.2	Characterisation of q -critical graphs of order n	140
7.3	Computing p -stable graphs of order n	142
7.4	Extremal stability and criticality values	148
7.4.1	The largest q for which an order n graph can be q -critical	148
7.4.2	The largest p for which an order n graph can be p -stable	149
7.5	Special graph classes	159
7.5.1	Complete and complete bipartite graphs	159
7.5.2	Paths and cycles	160
7.5.3	Dumbbells	163
7.6	Chapter summary	164

This chapter is concerned with establishing threshold information on the number of edge removals from a graph before increasing its secure domination number. The chapter opens in §7.1 with formal descriptions of the notions of criticality and stability which measure respectively the smallest number of arbitrary edges whose deletion necessarily increases the secure domination number of a graph, and the largest number of arbitrary edges whose deletion necessarily does not increase the secure domination number of a graph. An inductive characterisation of q -critical graphs follows in §7.2 and this characterisation is used to derive an algorithm for computing all q -critical graphs of small order. Similar results are established in §7.3 for the notion of stability, although the problem of characterising 0-stable graphs of order n remains open. Section 7.4 is concerned with determining the largest values of p and q for which a graph of order n is p -stable and q -critical, while §7.5 is dedicated to determining the exact values for p and q for which members of various special infinite classes of graphs are p -stable and q -critical. The chapter closes with a brief chapter summary in §7.6.

7.1 The notions of p -stability and q -criticality

A graph G is q -critical if the *smallest* arbitrary subset of edges whose removal from G necessarily increases the secure domination number, has cardinality q . Being able to determine the value of

q for which a given graph G is q -critical is important from an application point of view, because this value may be seen as a robustness threshold in the sense that the failure of any $q - 1$ edges in G results in a graph that can still be dominated securely by $\gamma_s(G)$ guards, but this is not true for the failure of *some* set of q edges in G . In terms of the cost functions of Chapter 6, a graph G is q -critical if $c_q(G) > 0$, but $c_{q-1}(G) = 0$.

A graph G is p -stable if the *largest* subset of arbitrary edges whose removal from G necessarily does *not* increase the secure domination number of the resulting graph, has cardinality p . Stability and criticality are therefore dual notions, and being able to determine the value of p for which a given graph G is p -stable is important in the same generic application as above, where this value may be seen as a robustness threshold in the sense that the failure of any $p + 1$ edges in G results in a graph that cannot be dominated securely by $\gamma_s(G)$ guards, but this is not true for the failure of *some* set of p edges in G . In terms of the cost functions of Chapter 6, a graph G is p -stable if $C_p(G) = 0$, but $C_{p+1}(G) > 0$.

The *edge-removal metagraph* of a graph G of size m is a graph whose nodes represent the non-isomorphic members of $G - qe$ for all $q = 0, 1, \dots, m$. The nodes of this edge-removal metagraph are arranged in $m + 1$ levels numbered $0, 1, \dots, m$ and the nodes on level q are the members of $G - qe$. A node H on level $q - 1$ of this metagraph is joined to a node H' on level q if H' can be obtained by removing one edge from H , for any $q = 1, 2, \dots, m$. The only node on level 0 of the edge-removal metagraph of some graph G corresponds to G itself, while the only node on level m corresponds to the edgeless graph of the same order as G . The cost sequences $\mathbf{c}(G)$ and $\mathbf{C}(G)$, introduced in Chapter 6, can be determined easily from the edge-removal metagraph of G . The edge-removal metagraph of the complete graph \mathcal{K}_n is of particular interest, because it contains nodes corresponding to all the non-isomorphic graphs of order n . The edge-removal metagraph of \mathcal{K}_4 is shown as an example in Figure 7.1.

The following result shows that as one moves down in the edge-removal metagraph of a graph, both the criticality and stability values of the graphs encountered are non-increasing, provided that the value of the secure domination number does not increase.

Lemma 7.1

- (a) Let G_1 be a q_1 -critical graph of size at least $q_1 > 0$. Then every graph $G_2 \in G_1 - 1e$ for which $\gamma_s(G_1) = \gamma_s(G_2)$ is q_2 -critical for some $q_2 < q_1$.
- (b) Let G_1 be a p_1 -stable graph of size at least $p_1 > 0$. Then there exists a p_2 -stable graph $G_2 \in G_1 - 1e$ for some $p_2 < p_1$.

Proof: (a) By contradiction. Let G_1 be a q_1 -critical graph for some integer $q_1 > 1$, but suppose, contrary to (a), that not all graphs $G_2 \in G_1 - 1e$ for which $\gamma_s(G_2) = \gamma_s(G_1)$ are r -critical for some $r \leq q_1 - 1$. Then there exists an s -critical graph $\tilde{H} \in G_1 - 1e$ such that $\gamma_s(\tilde{H}) = \gamma_s(G_1)$ for some $s > q_1 - 1$. But then

$$\begin{aligned} c_{q_1}(G_1) &= \min \gamma_s(G_1 - q_1 e) - \gamma_s(G_1) \\ &= \min \gamma_s((G_1 - 1e) - (q_1 - 1)e) - \gamma_s(G) \\ &\leq \min \gamma_s(\tilde{H} - (q_1 - 1)e) - \gamma_s(\tilde{H}) \\ &= c_{q_1-1}(\tilde{H}) \\ &= 0, \end{aligned}$$

contradicting the q_1 -criticality of G .

(b) By contradiction. Let G_1 be a p_1 -stable graph for some integer $p_1 > 0$. Note, therefore, that $\gamma_s(H) = \gamma_s(G_1)$ for all $H \in G_1 - 1e$. But suppose, contrary to (b), that every member of $G_1 - 1e$

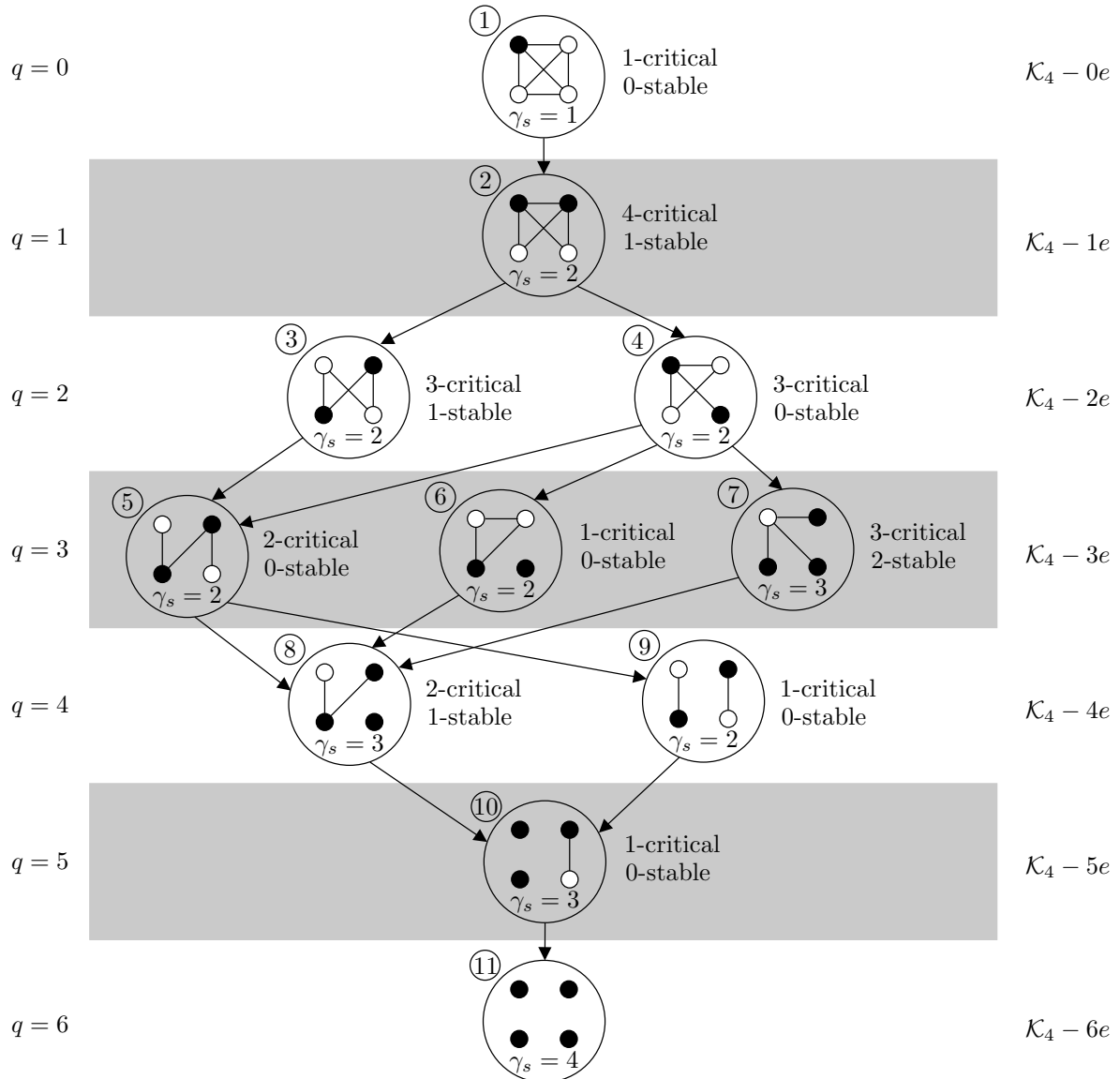


FIGURE 7.1: The edge-removal metagraph of the complete graph K_4 of order 4. The set $K_4 - qe$ is shown on level q of the graph for all $q = 0, \dots, 6$. Minimum secure dominating sets of the resulting graphs are denoted by solid vertices in each case. It follows that the sequences of cost functions of K_4 are $c(K_4) = 0, 1, 1, 1, 1, 2, 3$ and $C(K_4) = 0, 1, 1, 2, 2, 2, 3$.

is r -stable for some $r \geq p_1$. Then, there exists a graph $H' \in G_1 - 1e$ such that

$$\begin{aligned} C_{p_1+1}(G) &= \max \gamma_s(G_1 - (p_1 + 1)e) - \gamma_s(G_1) \\ &= \max \gamma_s((G_1 - 1e) - p_1e) - \gamma_s(G_1) \\ &= \max \gamma_s(H' - p_1e) - \gamma_s(H') \\ &= C_{p_1}(H') \\ &= 0, \end{aligned}$$

contradicting the p_1 -stability of G_1 . ■

Whereas it is possible to determine all p -stable and q -critical graphs of order n by constructing the edge-removal metagraph of the complete graph of order n and examining the secure domination numbers of the graphs at the nodes of this metagraph, this computational approach is only viable for very small values of n because of the excessive computation times associated with the construction of the edge-removal metagraph, as illustrated in Table 7.1.

n	2	3	4	5	6	7	8	9
Graphs	2	4	11	34	156	1 044	12 346	274 668
Time	$\ll 1$	$\ll 1$	$\ll 1$	< 1	1	16	1 664	1 069 220

TABLE 7.1: Computation times (in seconds) for constructing the edge-removal metagraphs of the complete graphs K_n of orders $n \in \{2, \dots, 9\}$ on a 3.4 GHz Intel(R) Core(TM) i7-3770 processor with 8 GiB RAM running in Ubuntu 12.04 using a C++ implementation of the edge-removal process illustrated in Figure 7.1.

These long computation times arise as a result of having to solve large numbers of instances of both the decision problem of determining whether two graphs are isomorphic and the decision problem associated with computing the secure domination number of a graph. It is clearly desirable to attempt to characterise the class of p -stable and q -critical graphs of order n for some fixed value of p or q so that the number of times that the above-mentioned **NP**-complete decision problems have to be resolved can be limited by not having to construct the entire edge-removal metagraph.

7.2 Characterisation of q -critical graphs of order n

Let \mathcal{Q}_n^q be the class of q -critical graphs of order n for some $n \in \mathbb{N}$ and some $q \in \{1, \dots, \binom{n}{2}\}$. Grobler and Mynhardt characterised the class \mathcal{Q}_n^1 , as described in §3.2.3. Using this characterisation, they derived a 4-step process for constructing the class \mathcal{Q}_n^1 . When this 4-step construction process is implemented for graphs of order 5, for example, then the seven nonisomorphic graphs in Figure 7.2 are obtained.

The following result is a characterisation of q -critical graphs in terms of $(q - 1)$ -critical graphs for $q = 2, 3, \dots$

Theorem 7.1 *A graph G of size at least $q > 1$ is q -critical if and only if*

- (a) *at least one graph $H \in G - 1e$ for which $\gamma_s(H) = \gamma_s(G)$ is $(q - 1)$ -critical, and*
- (b) *each graph $H \in G - 1e$ for which $\gamma_s(H) = \gamma_s(G)$ is r -critical for some $r \leq q - 1$.*

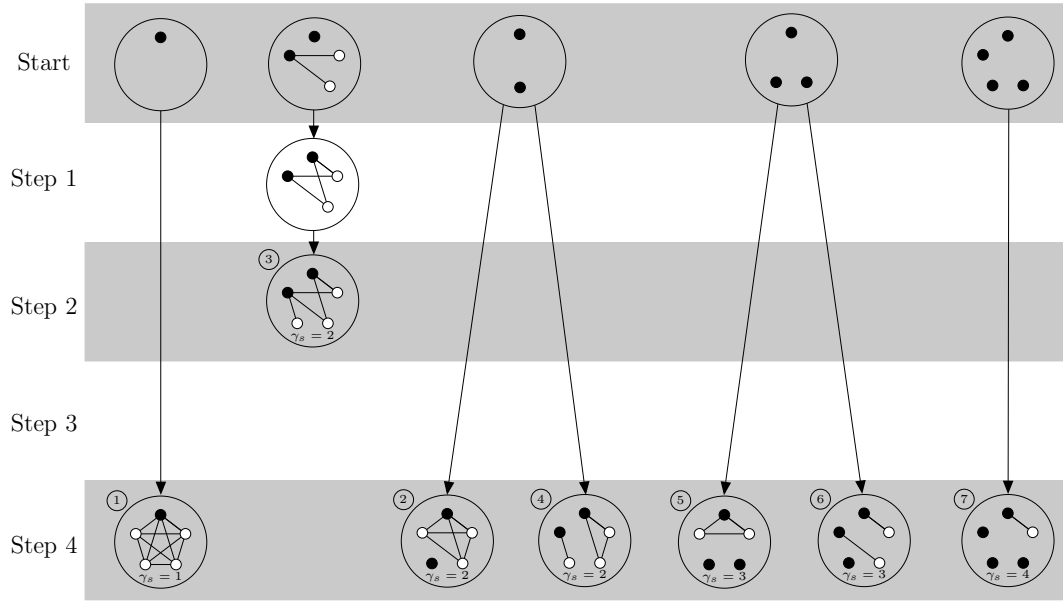


FIGURE 7.2: The 4-step process of Mynhardt and Grobler [55] for constructing the seven 1-critical graphs of order 5.

Proof: Let G be a q -critical graph for some integer $q > 1$. By Lemma 7.1(a), there is no r -critical graph $H \in G - 1e$ for any $r \geq p$ such that $\gamma_s(H) = \gamma_s(G)$, establishing the necessity of (b). The necessity of (a) is established by contradiction. Suppose, contrary to (a), that all graphs $H \in G - 1e$ for which $\gamma_s(H) = \gamma_s(G)$ are r -critical for some $r < q - 1$. Then there exists a graph $\hat{H} \in G - 1e$ with $\gamma_s(\hat{H}) = \gamma_s(G)$ such that

$$\begin{aligned}
 c_{q-1}(G) &= \min \gamma_s(G - (q-1)e) - \gamma_s(G) \\
 &= \min \gamma_s((G - 1e) - (q-2)e) - \gamma_s(G) \\
 &= \min \gamma_s(\hat{H} - (q-2)e) - \gamma_s(\hat{H}) \\
 &= c_{q-2}(\hat{H}) \\
 &\geq 1,
 \end{aligned}$$

contradicting the q -criticality of G and thereby establishing the necessity of (a).

The sufficiency of (a) and (b) are again established by contradiction. Suppose, to the contrary, that there exists a $(q-1)$ -critical member $H^* \in G - 1e$ for which $\gamma_s(H^*) = \gamma_s(G)$ and that each member of $G - 1e$ is r -critical for some $r \leq q-1$, but that G is not q -critical. It follows from Theorem 6.1 that $\gamma_s(H) \geq \gamma_s(G)$ for any graph $H \in G - 1e$. Moreover, $\min \gamma_s(H - (q-1)e) > \gamma_s(G)$ for any graph $H \in G - 1e$ such that $\gamma_s(H) = \gamma_s(G)$, since H is r -critical for some $r \leq q-1$. Therefore, $\gamma_s(H') > \gamma_s(G)$ for any graph $H' \in G - qe$, contradicting the supposition that G is not q -critical. ■

The result of Theorem 7.1 may be used to compute the class \mathcal{Q}_n^q inductively from the class \mathcal{Q}_n^{q-1} for any integer $n \geq 2$ and all permissible values of $q \geq 2$, using the above-mentioned 4-step construction process by Mynhardt and Grobler [55] for the class \mathcal{Q}_n^1 as base case. This inductive process is formalised in Algorithm 7.1. The algorithm commences by considering a graph $H \in \mathcal{Q}_n^{q-1}$ and proceeding to add a single edge $e^* \notin E(H)$ to H in Step 3, upon which the result of Theorem 7.1 is used to test whether or not $H + e^* \in \mathcal{Q}_n^q$. This process is repeated for each edge $e^* \notin E(H)$ and for each graph $H \in \mathcal{Q}_n^{q-1}$.

Algorithm 7.1: Computing the class of \mathcal{Q}_n^q of q -critical graphs of order n

Input : The class \mathcal{Q}_n^{q-1} of $(q-1)$ -critical graphs of order n .

Output: The class \mathcal{Q}_n^q of q critical graphs.

```

1 for each  $H \in \mathcal{Q}_n^{q-1}$  do
2   for each  $e^* \notin E(H)$  do
3     if  $q\text{-Critical}(H + e^*, q)$  then  $\mathcal{Q}_n^q \leftarrow \mathcal{Q}_n^q \cup \{H + e^*\};$ 
    
```

In Step 3 of Algorithm 7.1, another algorithm, Algorithm 7.2, is called to test whether $G = H + e^* \in \mathcal{Q}_n^q$. In Algorithm 7.2, each member of $G - 1e$ is examined. If a member $E \in G - 1e$ is found for which $\gamma_s(E) \neq \gamma_s(G)$, then $G \notin \mathcal{Q}_n^q$ by Theorem 7.1. Similarly, if a member $F \in \mathcal{Q}_n^{q'}$ is found for some $q' \geq q$, then $G \notin \mathcal{Q}_n^q$ by Theorem 7.1. If, however, no such graphs E or F are found, then $G \in \mathcal{Q}_n^q$ by Theorem 7.1, since $H \in \mathcal{Q}_n^{q-1}$.

Algorithm 7.2: $q\text{-Critical}(H + e^*, q)$

Input : A graph G and the value of q .

Output: A boolean value stating whether G is q -critical.

```

1 if  $G \in \mathcal{Q}_n^{q'}$  for some  $q' \leq q$  then
2   return [FALSE];
3 for each  $e \in E(G)$  do
4   if  $\gamma_s(G - e) = \gamma_s(G)$  and  $G - e \notin \mathcal{Q}_n^{q'}$  for some  $q' \leq q - 1$  then
5     return [FALSE];
6 return [TRUE];
    
```

The graph classes $\mathcal{Q}_5^1, \dots, \mathcal{Q}_5^6$ are shown in Figure 7.3. The 4-step construction of Grobler and Mynhardt [55] was used to compute the class \mathcal{Q}_n^1 in the first column of Figure 7.3 as base case (note that these graphs are exactly the seven graphs appearing in Figure 7.2). Thereafter, Algorithm 7.1 was used to compute the classes $\mathcal{Q}_5^2, \dots, \mathcal{Q}_5^6$ inductively.

Note that it is, in view of Theorem 7.1 and Algorithms 7.1–7.2, not necessary to construct the entire edge-removal metagraph of the complete graph of order n in order to determine the graph class \mathcal{Q}_n^q for a fixed value of q ; instead only the classes $\mathcal{Q}_n^1, \dots, \mathcal{Q}_n^q$ need be constructed inductively which, for values of q that are small compared to n , can be achieved in a fraction of the time required to construct the entire edge removal metagraph of \mathcal{K}_n .

The cardinalities of the nonempty graph classes \mathcal{Q}_n^q are listed in Table 7.2 for all $q \in \{0, 1, \dots, \binom{n}{2}\}$ and all $n \in \{2, 3, \dots, 9\}$.

7.3 Computing p -stable graphs of order n

Let \mathcal{S}_n^p be the class of non-isomorphic p -stable graphs of order $n \geq 2$. This section opens with a characterisation of the graph class \mathcal{S}_n^p in terms of the class \mathcal{S}_n^{p-1} for any natural number p .

Theorem 7.2 *A graph G of size at least $p > 0$ is p -stable if and only if*

- (a) $\gamma_s(H) = \gamma_s(G)$ for each $H \in G - 1e$ and
- (b) *each member of $G - 1e$ is r -stable for some $r \geq p - 1$ and at least one member of $G - 1e$ is $(p - 1)$ -stable.*

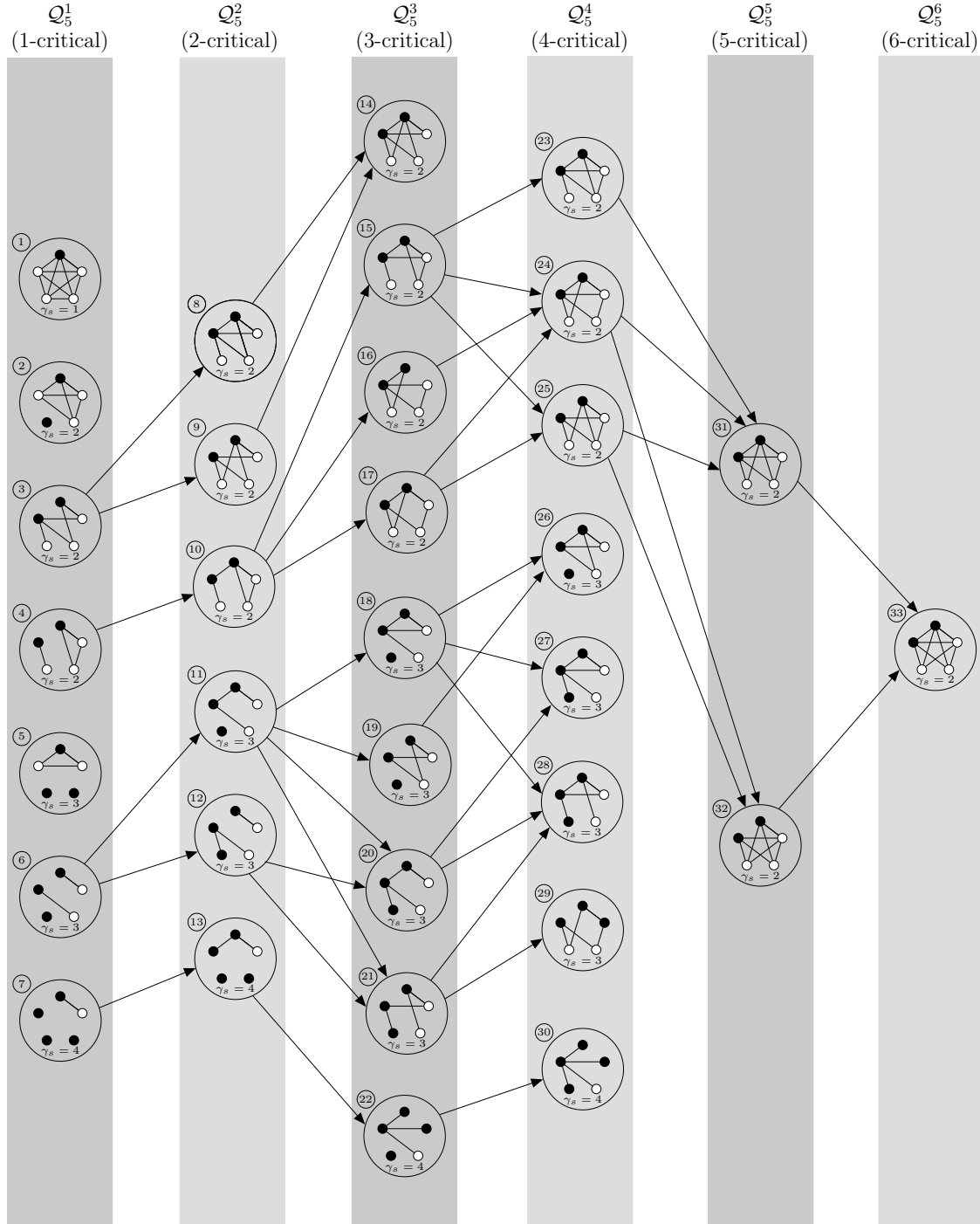


FIGURE 7.3: The graph classes Q_5^1, \dots, Q_5^6 . Minimum secure dominating sets are denoted by solid vertices in each case. An arrow from a member of $G - 1e$ to G denotes the q -criticality certificate relationship (in terms of graphs that are $(q - 1)$ -critical) described in Theorem 7.1(b).

$n \rightarrow$	Number of q -critical graphs of order n							
	2	3	4	5	6	7	8	9
$ Q_n^1 $	1	2	4	7	14	26	52	104
$ Q_n^2 $		1	2	6	18	50	141	394
$ Q_n^3 $			3	9	32	111	428	1 514
$ Q_n^4 $			1	8	34	165	910	4 424
$ Q_n^5 $				2	28	199	1 484	10 587
$ Q_n^6 $				1	18	195	1 875	20 144
$ Q_n^7 $					8	153	2 010	30 849
$ Q_n^8 $					2	93	1 847	38 831
$ Q_n^9 $					1	37	1 520	41 620
$ Q_n^{10} $						10	1 088	38 341
$ Q_n^{11} $						3	627	30 962
$ Q_n^{12} $						1	260	22 864
$ Q_n^{13} $							76	15 934
$ Q_n^{14} $							19	10 053
$ Q_n^{15} $							5	5 222
$ Q_n^{16} $							2	2 048
$ Q_n^{17} $							1	585
$ Q_n^{18} $								138
$ Q_n^{19} $								34
$ Q_n^{20} $								11
$ Q_n^{21} $								5
$ Q_n^{22} $								2
$ Q_n^{23} $								1
Total	1	3	10	33	155	1 043	12 345	274 667
Time	$\ll 1$	$\ll 1$	< 1	2	23	531	27 208	1 069 220

TABLE 7.2: Cardinalities of the nonempty graph classes $Q_n^1, \dots, Q_n^{\binom{n}{2}}$ for $n \in \{2, \dots, 9\}$ as computed on a 3.4 GHz Intel(R) Core(TM) i7-3770 processor with 8 GiB RAM running in Ubuntu 12.04 and using a C++ implementation of Algorithms 7.1–7.2 in conjunction with the Boost graph library [89] for graph isomorphism testing. The computation times, shown in the last row, are measured in seconds and represent the total time required to determine all the graph classes $Q_n^2, Q_n^3, Q_n^4, \dots$ from the graph class Q_n^1 .

Proof: Let G be a p -stable graph for some $p > 0$. Then $\gamma_s(G) = \gamma_s(H)$ for all graphs $H \in G - 1e$, implying the necessity of (a). The necessity of (b) is established by contradiction. Suppose, contrary to (b), that there exists an r -stable graph $\hat{H} \in G - 1e$ for some $r < p - 1$. Then,

$$\begin{aligned}
 C_p(G) &= \max \gamma_s(G - pe) - \gamma_s(G) \\
 &= \max \gamma_s((G - 1e) - (p - 1)e) - \gamma_s(G) \\
 &\geq \max \gamma_s(\hat{H} - (p - 1)e) - \gamma_s(\hat{H}) \\
 &= C_{p-1}(\hat{H}) \\
 &\geq 1,
 \end{aligned}$$

contradicting the p -stability of G . This shows that all graphs in the class $G - 1e$ are r -stable for some $r \geq p - 1$. Note, however, that by Lemma 7.1(b) there exists an r -stable graph $H' \in G - 1e$ for some $r < p$. Clearly, $r = p - 1$ in this case, thereby establishing the necessity of (b).

For the sufficiency of (a) and (b), suppose, to the contrary, that at least one member $H^* \in G - 1e$ for which $\gamma_s(H^*) = \gamma_s(G)$ is $(p - 1)$ -stable and that each member $H \in G - 1e$ for which $\gamma_s(H) = \gamma_s(G)$ is r -stable for some $r \geq p - 1$, but that G is not p -stable. Then $\max \gamma_s(G - pe) = \max \gamma_s(H - (p - 1)e) = \gamma_s(H) = \gamma_s(G)$, where the maximum is taken over all graphs $H \in G - 1e$ for which $\gamma_s(H) = \gamma_s(G)$. But $\max \gamma_s(G - (p + 1)e) \geq \gamma_s(H^* - pe) > \gamma_s(H^*) = \gamma_s(G)$, contradicting the supposition that G is not p -stable. ■

If the class \mathcal{S}_n^0 can be characterised and constructed, then the result of Theorem 7.2 may be used to compute the class \mathcal{S}_n^p inductively from the class \mathcal{S}_n^{p-1} for all $p \in \mathbb{N}$ and all $n \geq 2$, using the class \mathcal{S}_n^0 as base case. The following open problem is therefore posed.

Problem 7.1 *Characterise the class \mathcal{S}_n^0 of 0-stable graphs of order n .*

While the above problem seems hard, it is easy to prove that \mathcal{S}_n^0 is nonempty for all $n \geq 2$. The next result follows immediately from Theorem 4.1 since the complete graph \mathcal{K}_n is the only graph of order n with secure domination number 1 by Theorem 4.1.

Theorem 7.3 $\mathcal{K}_n \in \mathcal{S}_n^0$ for all $n \geq 2$. ■

In fact, it follows from Theorem 7.3 that any graph of order n which contains a nontrivial, complete component is a member of \mathcal{S}_n^0 . The seven graphs in Figure 7.4 are, for example, all members of the graph class \mathcal{S}_5^0 by Theorem 7.3.

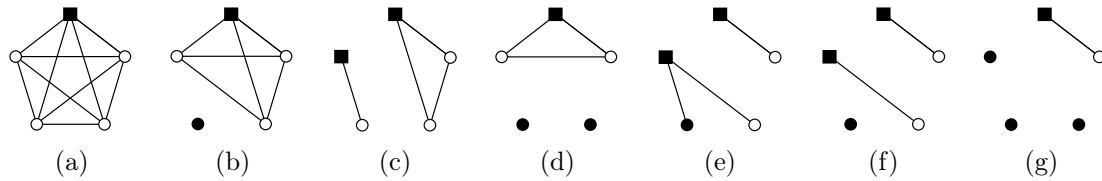


FIGURE 7.4: A subset of the graph class \mathcal{S}_5^0 . Minimum secure dominating sets are denoted by solid vertices in each case. Square solid vertices denote members of the minimum secure dominating set which have external private neighbours.

Furthermore, each member of the graph class \mathcal{S}_n^0 necessarily has the following interesting property.

Theorem 7.4 *If $G \in \mathcal{S}_n^0$ for some $n \geq 2$, then some vertex in every minimum secure dominating set X of G necessarily has private neighbours external to X .*

Proof: Suppose G is 0-stable, but suppose, to the contrary, that G possesses a minimum secure dominating set X in which no vertex has private neighbours external to X . Let $e = uv$ be any edge of G . Then there are three cases to consider:

Case i: $u, v \in X$. In this case, clearly, X remains a secure dominating set of $G - e$, showing that $\gamma_s(G - e) \leq |X| = \gamma_s(G)$. But since $\gamma_s(G - e) \geq \gamma_s(G)$ by Theorem 6.1, it follows that $\gamma_s(G - e) = \gamma_s(G)$.

Case ii: $u, v \notin X$. Since u and v are not external private neighbours of any vertex in X , it follows from Theorem 3.6 that they are not uniquely defended by a common vertex in X . Hence X remains a secure dominating set of $G - e$, and so $\gamma_s(G - e) = \gamma_s(G)$, as above.

Case iii: $u \in X$ and $v \notin X$. Since no vertex outside X is an external private neighbour of any vertex in X , v has another neighbour in X which does not defend any vertex outside X uniquely. Hence X remains a secure dominating set of $G - e$, and so $\gamma_s(G - e) = \gamma_s(G)$ yet again.

In all of the above cases, removal of the arbitrary edge e from G does not increase the domination number of the resulting graph, contradicting the 0-stability of G . ■

The property in Theorem 7.4 is, however, not sufficient to characterise the graph class \mathcal{S}_n^0 , as illustrated by the counter example in Figure 7.5.

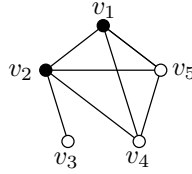


FIGURE 7.5: A minimum secure dominating set $X = \{v_1, v_2\}$ for a graph of order 5 which is not in \mathcal{S}_5^0 . Yet v_3 is a private neighbour of v_2 external to X .

If Problem 7.1 can be solved and if the resulting characterisation can be used to derive an iterative or recursive construction for the graph class \mathcal{S}_n^0 , then the inductive process of computing the graph classes $\mathcal{S}_n^1, \mathcal{S}_n^2, \mathcal{S}_n^3, \dots$ from \mathcal{S}_n^0 can be achieved by Algorithm 7.3. The algorithm commences by systematically considering each graph $H \in \mathcal{S}_n^{p-1}$ and proceeds to add a single edge $e \notin E(H)$ to H in Step 3, upon which the result of Theorem 7.2 is used to test whether $H + e$ is, in fact, p -stable. If $H + e$ is p -stable, it is included in the class \mathcal{S}_n^p . This process is repeated for each edge $e \notin E(H)$.

Algorithm 7.3: The class \mathcal{S}_n^p of p -stable graphs of order n for $p > 0$

Input : The classes $\mathcal{S}_n^0, \dots, \mathcal{S}_n^{p-1}$.

Output: The class \mathcal{S}_n^p of p -stable graphs of order n .

```

1 for each  $H \in \mathcal{S}_n^{p-1}$  do
2   for each  $e \notin E(H)$  do
3     if  $p\text{-Stable}(H + e, p)$  then  $\mathcal{S}_n^p \leftarrow \mathcal{S}_n^p \cup \{H + e\};$ 

```

In Step 3 of Algorithm 7.3, Algorithm 7.4 is called to test whether the graph $G = H + e$ is p -stable. In Algorithm 7.4, each member of $G - 1e$ ($e \in E(G)$) is examined. If a member I is found for which $\gamma_s(I) \neq \gamma_s(G)$, then G is not p -stable. Furthermore, if $G - e \notin \mathcal{S}_n^r$ for $r \geq p - 1$, then again G is not p -stable by Theorem 7.2.

Algorithm 7.4: $p\text{-Stable}(G, p)$

Input : A graph G of order n , a natural number p and the graph classes $\mathcal{S}_n^0, \dots, \mathcal{S}_n^{p-1}$.

Output: TRUE if $G \in \mathcal{S}_n^p$, or FALSE otherwise.

```

1 if  $G \in \mathcal{S}_n^q$  for some  $q \leq p$  then
2   return [FALSE];
3 for each  $e \in E(G)$  do
4   if  $\gamma_s(G - e) \neq \gamma_s(G)$  or  $G - e \in \mathcal{S}_n^q$  for some  $q < p - 1$  then
5     return [FALSE];
6 return [TRUE];

```

The graph classes $\mathcal{S}_n^0, \mathcal{S}_n^1, \mathcal{S}_n^2$ and \mathcal{S}_n^3 are shown as an example in Figure 7.6. The base case class \mathcal{S}_5^0 shown in the outer layer of Figure 7.6, was found manually. Thereafter, Algorithm 7.3 was used to compute the classes $\mathcal{S}_5^1, \mathcal{S}_5^2$ and \mathcal{S}_5^3 inductively. These classes are shown in the inner layers of the figure.

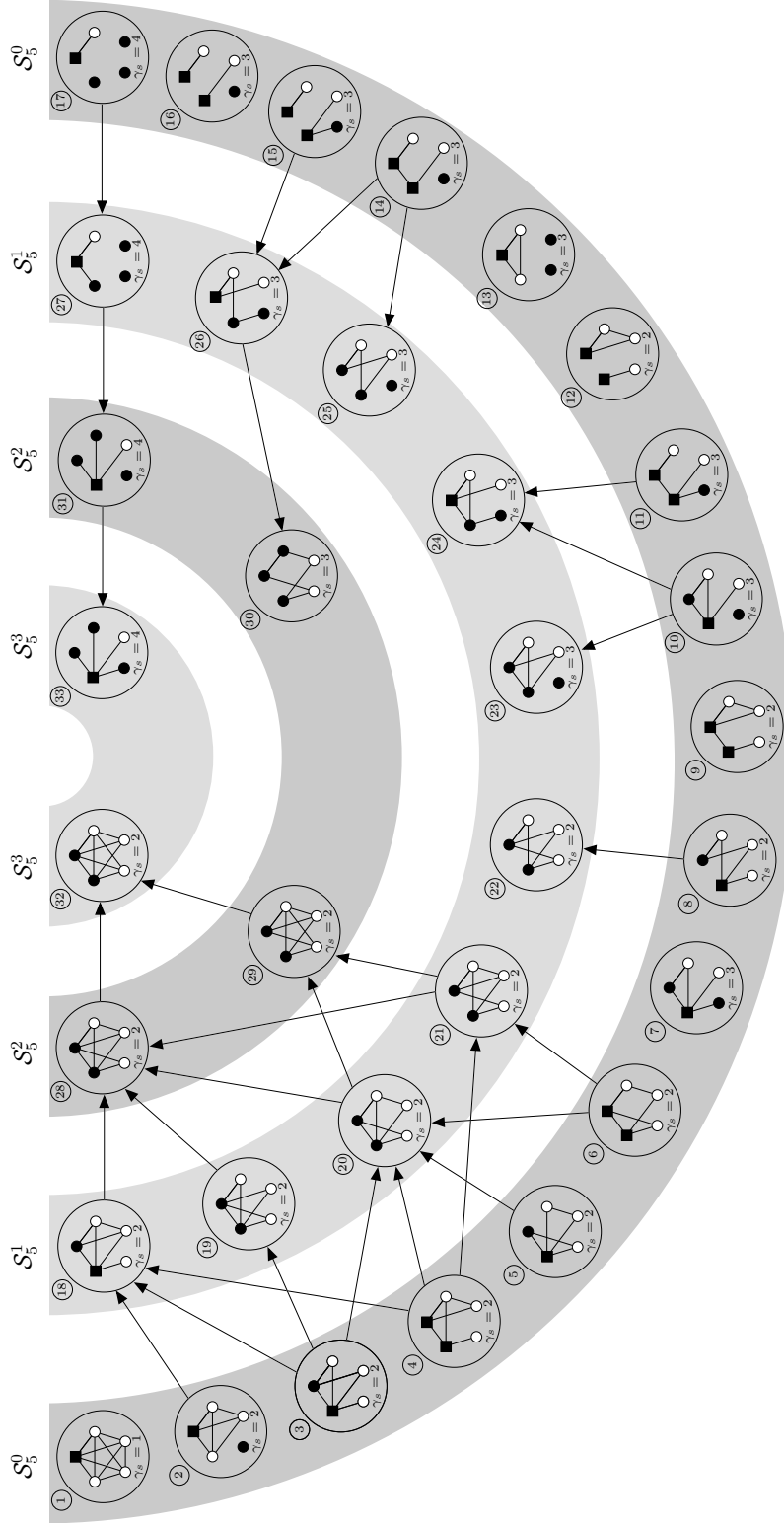


FIGURE 7.6: The graph classes S_5^0, S_5^1, S_5^2 and S_5^3 . Minimum secure dominating sets are denoted by solid vertices in each case. Square vertices in these minimum secure dominating sets have external private neighbours. An arrow from a member of $G - 1e$ to G denotes the p -stability certificate relationship (in terms of graphs that are $(p - 1)$ -stable) described in Theorem 7.2(b).

The cardinalities of the nonempty graph classes \mathcal{S}_n^p are listed in Table 7.3 for all $p \in \{0, 1, \dots, \binom{n}{2}\}$ and all $n \in \{2, 3, \dots, 9\}$.

$n \rightarrow$	2	3	4	5	6	7	8	9
$ \mathcal{S}_n^0 $	1	2	6	17	81	514	5 460	107 794
$ \mathcal{S}_n^1 $		1	3	10	51	355	4 205	94 106
$ \mathcal{S}_n^2 $			1	4	16	136	2 050	52 502
$ \mathcal{S}_n^3 $				2	5	32	551	16 923
$ \mathcal{S}_n^4 $					2	5	70	3 081
$ \mathcal{S}_n^5 $						1	6	245
$ \mathcal{S}_n^6 $							3	13
$ \mathcal{S}_n^7 $								3
Total	1	3	10	33	155	1 043	12 345	274 667
Time		$\ll 1$	< 1	1	15	374	15 895	1 069 220

TABLE 7.3: Cardinalities of the nonempty graphs classes $\mathcal{S}_n^1, \dots, \mathcal{S}_n^{\binom{n}{2}}$ of orders $n \in \{2, \dots, 9\}$ computed by a C++ implementation of Algorithms 7.3–7.4 on a 3.4 GHz Intel(R) Core(TM) i7-3770 processor with 8 GiB RAM running in Ubuntu 12.04. The Boost graph library [89] was used for isomorphism testing. The classes $\mathcal{S}_2^0, \mathcal{S}_3^0, \dots, \mathcal{S}_9^0$ were found by brute force and the constructions of these classes were not included in the time measurements in the last row of the table (which are measured in seconds).

7.4 Extremal stability and criticality values

The inductive process described in §7.2, together with the fact that the class \mathcal{Q}_n^1 is nonempty for all $n \geq 2$, leads to the following observation.

Observation 7.1 *For any integer $n \geq 2$ there exists a natural number $\Omega_n \leq \binom{n}{2}$ such that*

$$\mathcal{Q}_n^1, \dots, \mathcal{Q}_n^{\Omega_n} \neq \emptyset \quad \text{and} \quad \mathcal{Q}_n^{\Omega_n+1}, \dots, \mathcal{Q}_n^{\binom{n}{2}} = \emptyset. \quad \blacksquare$$

Similarly, it follows from Theorem 7.3 that $\mathcal{S}_n^0 \neq \emptyset$ for all $n \geq 2$. Hence Theorem 7.2 therefore implies the following result.

Observation 7.2 *For any integer $n \geq 2$ there exists a natural number $\omega_n \leq \binom{n}{2}$ such that*

$$\mathcal{S}_n^0, \dots, \mathcal{S}_n^{\omega_n} \neq \emptyset \quad \text{and} \quad \mathcal{S}_n^{\omega_n+1}, \dots, \mathcal{S}_n^{\binom{n}{2}} = \emptyset. \quad \blacksquare$$

7.4.1 The largest q for which an order n graph can be q -critical

With Ω_n as defined in Observation 7.1, it is easily established that $\Omega_2 = 1$ and $\Omega_3 = 2$. Moreover, $\Omega_4 = 4$ from Figure 7.1 and $\Omega_5 = 6$ from Figure 7.3. Furthermore, $\Omega_6 = 9$, $\Omega_7 = 12$, $\Omega_8 = 17$ and $\Omega_9 = 23$ from Table 7.2. Determining the value of Ω_n for an arbitrary value of n , however, seems to be a hard problem. It is nevertheless possible to establish the following lower bound on Ω_n .

Theorem 7.5 $\Omega_n \geq \binom{n}{2} - 2n + 5$ for all $n \geq 7$.

Proof: A graph of order n that is q -critical for some $q \geq \binom{n}{2} - 2n + 5$ is exhibited. Since the graphs $\Phi(n-1, 1, 0, 0)$ and $\Phi(2, 1, 0, n-3)$ in Figure 4.7(a)–(b) are spanning subgraphs of $\mathcal{K}_n - e$ and $\mathcal{K}_{2,n-2} - e$, respectively, it follows from Theorem 4.2 that $\gamma_s(\mathcal{K}_n - e) = \gamma_s(\mathcal{K}_{2,n-2} - e) = 2$. The fact that $\mathcal{K}_{2,n-2} - e$ is a subgraph of $\mathcal{K}_n - e$ shows, however, that it is possible to remove $\binom{n}{2} - 1 - [2(n-2) - 1] = \binom{n}{2} - 2n + 4$ edges from the near complete graph $\mathcal{K}_n - e$ without increasing the secure domination number of the resulting graph, and so $\mathcal{K}_n - e$ is not q -critical for any $q \leq \binom{n}{2} - 2n + 4$. ■

The result of Theorem 7.5 and the circumstantial numerical evidence in Table 7.2 suggests the following conjecture.

Conjecture 7.1 $\Omega_n = \binom{n}{2} - 2n + 5$ for all $n \geq 10$.

7.4.2 The largest p for which an order n graph can be p -stable

With ω_n as defined in Observation 7.2, it follows from Table 7.3 that $\omega_n = n - 2$ for all $n \in \{2, \dots, 9\}$. The author suspects that this linear functional relation holds in general; hence the following conjecture.

Conjecture 7.2 $\omega_n = n - 2$ for all $n \geq 2$.

Conjecture 7.2 is further substantiated by establishing linear bounds on ω_n .

Theorem 7.6 Suppose u and v are two adjacent vertices of a graph G . Then G is p -stable for some $p \leq \deg(u) + \deg(v) - 2$.

Proof: Let u and v be two adjacent vertices in G , and let H be the graph obtained by removing all the edges from G that are incident to both u and v , except for the edge uv . Then H contains $\deg(u) + \deg(v) - 2$ edges fewer than G and one component more than G . Furthermore, one of these components is isomorphic to the complete graph \mathcal{K}_2 , which is 0-stable. Hence it is possible to increase the secure domination number of G by removing more than $\deg(u) + \deg(v) - 2$ edges, and so G is not p -stable for any $p \geq \deg(u) + \deg(v) - 1$. ■

The following three consequences immediately follow from Theorem 7.6.

Corollary 7.1 A connected graph with maximum degree Δ is p -stable for some $p \leq 2(\Delta - 1)$.

Corollary 7.2 A connected graph with minimum degree δ and maximum degree $\Delta = n - 1$ is p -stable for some $p \leq \Delta + \delta - 3$.

Corollary 7.3 $\omega_n \leq 2n - 4$ for all $n \geq 2$.

When further restrictions are placed on the graph, then improvements on the above bounds, such as the following, may be obtained.

Theorem 7.7 *If G contains a vertex v such that $\gamma_s(G - v) \geq \gamma_s(G)$, then G is p -stable for some $p \leq \deg(v) - 1 \leq \Delta - 1$.*

Proof: Let v be a vertex of G such that $\gamma_s(G - v) \geq \gamma_s(G)$ and let H be the graph obtained by removing all edges from G that are incident to v . Then $\gamma_s(H) = \gamma_s(G - v) + 1 \geq \gamma_s(G) + 1 > \gamma_s(G)$. Hence it is possible to increase the secure domination number of G by removing $\deg(v)$ edges from G , and so G is only p -stable for some $p \leq \deg(v) - 1 \leq \Delta - 1$. ■

The following corollaries are immediate consequences of Theorem 7.7.

Corollary 7.4 *If G contains a vertex v such that $\gamma_s(G - v) \geq \gamma_s(G)$, then G is p -stable for some $p \leq n - 2$.*

Corollary 7.5 *Let G be an $(n - 2)$ -stable graph. Then $\gamma_s(G - v) < \gamma_s(G)$ for any non-universal vertex v of G .*

Although a vertex v for which $\gamma_s(G - v) \geq \gamma_s(G)$ may often be found in a graph G , this requirement is not necessary for the result of Corollary 7.4. For example, the complete bipartite graph $\mathcal{K}_{4,4}$ has no such vertex, yet $\mathcal{K}_{4,4}$ is 6-stable, as is shown next.

Theorem 7.8 $\mathcal{K}_{4,4} \in \mathcal{S}_8^6$.

Proof: Suppose V_1 and V_2 are the partite sets of $\mathcal{K}_{4,4}$ and let H be a subgraph of $\mathcal{K}_{4,4}$ that is isomorphic to a member of $\mathcal{K}_{4,4} - 6e$. Since $H \subseteq \mathcal{K}_{4,4}$ and $\gamma_s(\mathcal{K}_{4,4}) = 4$ by Proposition 3.7,

$$\gamma_s(H) \geq 4. \quad (7.1)$$

It is shown, by considering a number of cases, that, in fact, $\gamma_s(H) = 4$.

The largest subgraph of $\mathcal{K}_{4,4}$ containing a vertex $v \in V_1$ with $d \in \{0, 1, 2, 3, 4\}$ private neighbours $u_1, \dots, u_d \in V_2$ is obtained by joining v to all vertices in V_2 and joining each of the three vertices in $V_1 - \{v\}$ to each of the $4 - d$ vertices in $V_2 - \{u_1, \dots, u_d\}$. Because this subgraph has size $4 + 3(4 - d) = 16 - 3d$ and H has size 10, it follows that $d \leq 2$ (i.e. H has no vertex with more than two private neighbours). Moreover, if H has a vertex with two private neighbours, then it has exactly one such vertex, v (say), and H is necessarily isomorphic to the graph in Figure 7.7(a). In this case H has the secure dominating set of cardinality 4 indicated by the solid vertices in the figure, showing that $\gamma_s(H) = 4$ by (7.1).

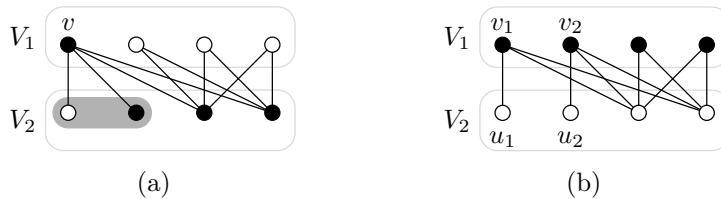


FIGURE 7.7: The graph H containing (a) a vertex with two private neighbours, and (b) two vertices, each with exactly one private neighbour.

Suppose, therefore, that no vertex in H has more than one private neighbour and let the number of vertices of H with exactly one private neighbour be $k \in \{0, 1, 2, 3, 4\}$. The largest subgraph

of $\mathcal{K}_{4,4}$ containing k vertices $v_1, \dots, v_k \in V_1$, each with one private neighbour, is obtained by joining v_i to a private neighbour $u_i \in V_2$ for each $i = 1, \dots, k$ and by joining each vertex in $V_1 - \{v_1, \dots, v_k\}$ to all vertices in $V_2 - \{u_1, \dots, u_k\}$. Because this subgraph of $\mathcal{K}_{4,4}$ has size $k + 4(4 - k) = 16 - 3k$ and H has size 10, it follows that $k \leq 2$. Furthermore, if H has exactly two vertices, each with exactly one private neighbour, then H is necessarily isomorphic to the graph in Figure 7.7(b). In this case H again has a secure dominating set of cardinality 4, as indicated by the solid vertices in the figure, showing that $\gamma_s(H) = 4$ by (7.1).

It may therefore be assumed that exactly one vertex of H , w (say), has at most one private neighbour and that the remaining vertices of H have no private neighbours. Furthermore, since it requires at least $4 + 3 = 7$ edge removals from $\mathcal{K}_{4,4}$ to isolate two vertices of the graph, H has at most one isolated vertex. Hence there are four final cases to consider:

Case i: H has one isolated vertex, x (say), and w has one private neighbour w' . In this case x and w' cannot be in the same partite set, because the largest subgraph satisfying these conditions has size 9, as shown in Figure 7.8(a). There is, however, only one subgraph of size 10 of $\mathcal{K}_{4,4}$ in which x and w' are in different partite sets, as shown in Figure 7.8(b). In this case H yet again has a secure dominating set of cardinality 4, as indicated by the solid vertices in the figure, showing that $\gamma_s(H) = 4$ by (7.1).

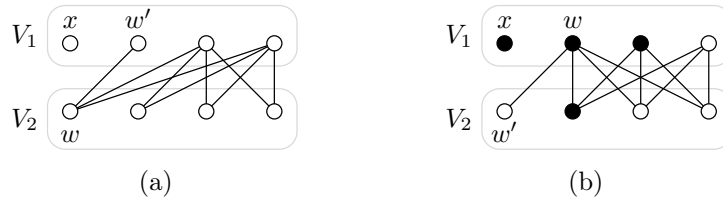


FIGURE 7.8: The graph H with one isolated vertex x , and one vertex w which has one private neighbour w' , where (a) x and w' are in the same partite set, and (b) x and w' are in the different partite sets.

Case ii: H has an isolated vertex, (x say), but no vertex with a private neighbour. Suppose $x \in V_1$ and let v_1, v_2 and v_3 be the other vertices in V_1 . Since $\deg(v_1) + \deg(v_2) + \deg(v_3) = 10$, it follows by the pigeonhole principle that at least one vertex in V_1 has degree 4; suppose this vertex is v_1 . Then, since $\deg(v_2) + \deg(v_3) = 6$, it follows by the pigeonhole principle that at least one of v_2 or v_3 has degree at least 3; suppose this vertex is v_2 and let the common neighbours of v_1 and v_2 be w_1, w_2 and w_3 . Then the remaining vertex w_4 in V_2 must be adjacent to either v_2 or v_3 , for otherwise it would be a private neighbour of v_1 . If w_4 is adjacent to v_2 , then H contains the subgraph in Figure 7.9(a), in which case v_3 must be adjacent to exactly two vertices in V_2 , w_i and w_j (say). But then H has the secure dominating set $\{x, v_1, w_i, w_j\}$ of cardinality 4. If, however, w_4 is adjacent to v_3 , then H contains the subgraph in Figure 7.9(b) which also has a secure dominating set of cardinality 4, as indicated by the solid vertices in the figure. In both cases, therefore, $\gamma_s(H) = 4$ by (7.1).

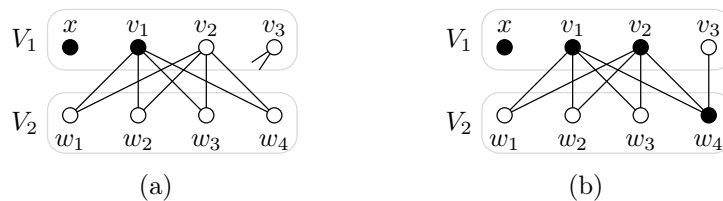


FIGURE 7.9: Subgraphs of H with an isolated vertex, (x say), but where no vertex has a private neighbour and where (a) v_3 is adjacent to w_i and w_j , and (b) v_3 is adjacent to w_4 .

Case iii: H has no isolated vertices, and w has one private neighbour w' . Suppose $w \in V_1$ and let v_1, v_2 and v_3 be the other vertices in V_1 . Since $\deg(w) + \deg(v_1) + \deg(v_2) + \deg(v_3) = 10$, it follows by the pigeonhole principle that at least two vertices in V_1 have degree at least 3. If w has degree at least 3, let v_1 be the another vertex in V_1 with degree at least 3. Then H contains the subgraph in Figure 7.10(a), in which case some vertex $u \in V_2$, which is adjacent to v_1 , must be adjacent to at least one other vertex in V_1 , v_3 (say). But then H has the secure dominating set $\{w, v_1, v_2, v_3\}$ of cardinality 4. If, however, w has degree 2, then two other vertices in V_1 , v_1 and v_2 (say), must have degree at least 3. Then H contains the subgraph in Figure 7.10(b) which also has a secure dominating set of cardinality 4, as indicated by the solid vertices in the figure. In both cases, therefore, $\gamma_s(H) = 4$ by (7.1).

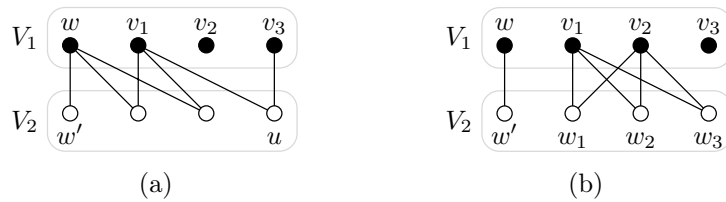


FIGURE 7.10: Subgraphs of H with no isolated vertices, where w has one private neighbour w' and where (a) w has degree at least 3, and (b) w has degree at most 2.

Case iv: H has no isolated vertices and no vertices with private neighbours. Let v_1, v_2, v_3 and v_4 be the vertices in V_1 . Since $\delta(H) \geq 2$, it follows by the pigeonhole principle that at least two vertices in V_1 , v_1 and v_2 (say), have degree 2. Furthermore, without loss of generality, either v_3 has degree 2 and v_4 has degree 4, or else both v_3 and v_4 have degree 3. If v_3 has degree 2 and v_4 has degree 4, then H contains the subgraph in Figure 7.11(a), in which case H has the secure dominating set $\{v_1, v_2, v_3, v_4\}$, because $N(v_1) \cup N(v_2) \cup N(v_3) = V_2$ in order to avoid private neighbours in H . In this case, therefore, $\gamma_s(H) = 4$ by (7.1).

Finally, if both v_3 and v_4 have degree 3, then $N(v_3) = N(v_4)$ or $N(v_3) \neq N(v_4)$. If $N(v_3) = N(v_4)$, then there is a vertex $u \in V_2$ that is adjacent to neither v_3 nor v_4 , in order to avoid private neighbours in H . Since $\delta(H) \geq 2$, u is adjacent to v_1 and v_2 , as shown in Figure 7.11(b), in which case H has the secure dominating set $\{v_1, v_2, v_3, v_4\}$ of cardinality 4, showing that $\gamma_s(H) = 4$ by (7.1). If, however, $N(v_3) \neq N(v_4)$, then some vertex $u \in V_2$ is adjacent to v_3 but not to v_4 . Since $\delta(H) \geq 2$, u is adjacent to at least one other vertex in V_1 . Assume, without loss of generality, that u is adjacent to v_1 . Then H contains the subgraph in Figure 7.11(c), in which case H has the secure dominating set $\{v_1, v_2, v_3, v_4\}$ of cardinality 4, showing that $\gamma_s(H) = 4$ by (7.1).

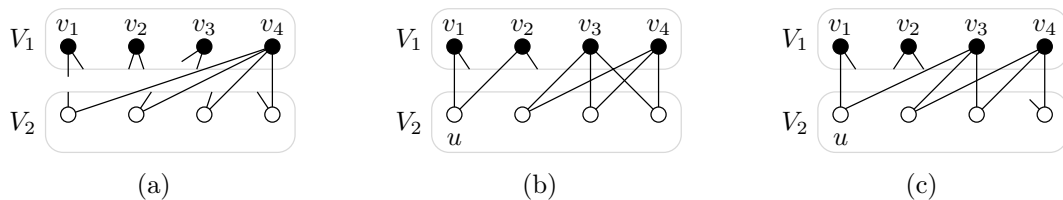


FIGURE 7.11: The graph H with no isolated vertices and no vertices with private neighbours, where (a) $\deg(v_3) = 2$ and $\deg(v_4) = 4$, (b) $\deg(v_3) = \deg(v_4) = 3$ and $N(v_3) = N(v_4)$, and (c) $\deg(v_3) = \deg(v_4) = 3$ and $N(v_3) \neq N(v_4)$.

From the above cases it is concluded that $\gamma_s(H) = 4$ and hence that $\mathcal{K}_{4,4}$ is p -stable for some $p \geq 6$. The subgraph $2\mathcal{K}_1 \cup \mathcal{K}_{3,3}$ of size $16 - 7 = 9$ of $\mathcal{K}_{4,4}$, however, has secure domination number $2 + 3 = 5$ by Proposition 3.7, showing that $\mathcal{K}_{4,4}$ is not p -stable for any $p \geq 7$. ■

The result of Theorem 7.8 is an isolated instance. The graph in Figure 7.12, for example, is a member of $\mathcal{K}_{r,r} - 2(r-1)e$ which has no secure dominating set of cardinality 4. This shows that the proof of case i in Theorem 7.8 cannot be generalised from $\mathcal{K}_{4,4}$ to $\mathcal{K}_{r,r}$ and hence that $\mathcal{K}_{r,r} \notin \mathcal{S}_n^{n-2}$ for $n = 2r$, where $r > 4$ is an integer.

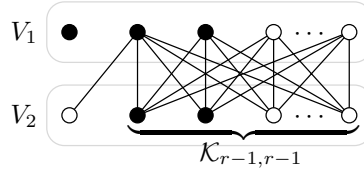


FIGURE 7.12: A member of $\mathcal{K}_{r,r} - 2(r-1)e$ which has no secure dominating set of cardinality 4. A minimum secure dominating set is indicated by solid vertices.

The following result is a similarly isolated case.

Theorem 7.9 $\mathcal{K}_{3,3,3} \in \mathcal{S}_9^7$.

Proof: Suppose V_1, V_2 and V_3 are the partite sets of $\mathcal{K}_{3,3,3}$ and let H be a subgraph of $\mathcal{K}_{3,3,3}$ that is isomorphic to a member of $\mathcal{K}_{3,3,3} - 7e$. Since $H \subseteq \mathcal{K}_{3,3,3}$ and $\gamma_s(\mathcal{K}_{3,3,3}) = 3$ by Proposition 3.8,

$$\gamma_s(H) \geq 3. \quad (7.2)$$

It is shown, by considering a number of cases, that, in fact, $\gamma_s(H) = 3$.

Let $V_1 = \{u_1, u_2, u_3\}$, $V_2 = \{v_1, v_2, v_3\}$ and $V_3 = \{w_1, w_2, w_3\}$. It follows from the pigeonhole principle that at least four vertices in H have degree at least 5. Furthermore, two vertices of the same partite set of H , $u_1, u_2 \in V_1$ (say), have degree at least 5. Two cases are considered.

Case i: $\Delta(H) = 6$. If both u_1 and u_2 have degree 6, then H contains the subgraph in Figure 7.13(a), in which case the partite set V_1 is a secure dominating set of H (of cardinality 3). Suppose then that only one vertex, u_2 (say), has degree 6, while u_1 has degree 5. Then H contains the subgraph in Figure 7.13(b), in which case the partite set V_1 is yet again a secure dominating set of H (of cardinality 3). In both cases, therefore, $\gamma_s(H) = 3$ by (7.2).

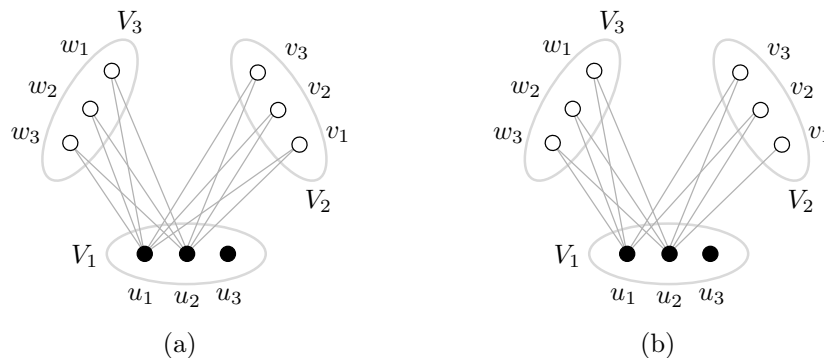


FIGURE 7.13: Subgraphs of H with $\Delta(H) = 6$, where (a) $\deg(u_1) = \deg(u_2) = 6$, with both u_1 and u_2 defending $V_1 \cup V_2$, and (b) $\deg(u_2) = 6$ and $\deg(u_1) = 5$, with u_2 defending v_1 , and u_1 defending $(V_2 - \{v_1\}) \cup V_3$. A minimum secure dominating set is indicated by the solid vertices in each case.

Case ii: $\Delta(H) \leq 5$. In this case at least two vertices, u_1 and u_2 (say), of V_1 have degree 5. It follows by the pigeonhole principle that $\delta(H) \geq 1$. However, if $\delta(H) = 1$, then H is isomorphic

to either the graph in Figure 7.14(a), the graph in Figure 7.14(b) or the graph in Figure 7.14(c). In all of these cases $\gamma_s(H) = 3$. Assume, therefore, that $\delta(H) \geq 2$. There are two subcases to consider:

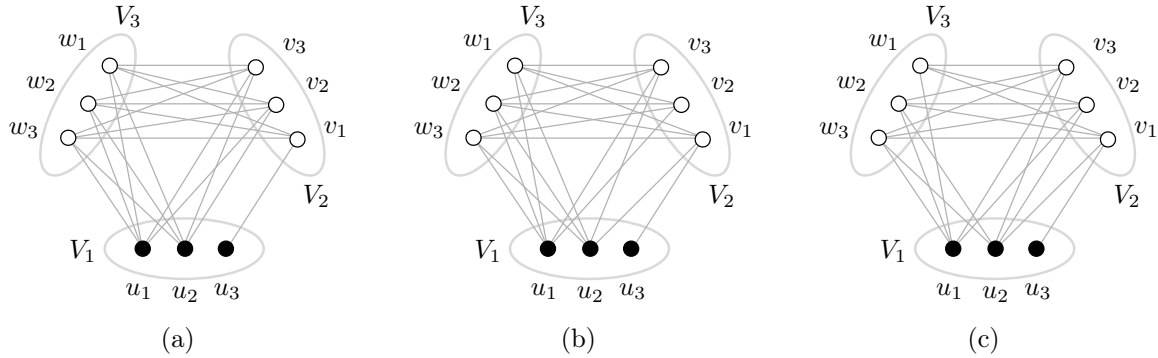


FIGURE 7.14: The graph H with $\Delta(H) = 5$ and $\delta(H) = 1$, with (a) $\deg(v_1) = 4$, where u_3 defends v_1 and u_1 defends $V_1 \cup (V_2 - \{v_1\})$, (b) $\deg(v_1) = 5$ and $\deg(v_2) = 4$, where u_1 defends v_3 and u_1 defends $V_1 \cup (V_2 - \{v_1\})$, and (c) $\deg(v_1) = 5$ and $\deg(w_1) = 4$, where u_3 defends v_1 and u_1 defends $V_1 \cup (V_2 - \{v_1\})$. A minimum secure dominating set is indicated by the solid vertices in each case.

Case ii(a): $N(u_1) = N(u_2)$. In this case exactly one vertex, v_1 (say), in $V_2 \cup V_3$ is adjacent to neither u_1 nor u_2 . If v_1 and u_3 are adjacent, then H contains the subgraph in Figure 7.15(a), in which case V_1 is a secure dominating set of H . Suppose therefore that v_1 and u_3 are not adjacent. Since v_1 is not adjacent to any vertex in $V_1 \cup V_2$ and u_3 is not adjacent to any vertex in $V_1 \cup \{v_1\}$, and since $\deg(v_1) + \deg(u_3) \geq 7$, it follows that u_3 and v_1 share at least two common neighbours, w_1 and w_2 (say), in V_3 , as shown in Figure 7.15(b). Furthermore, since $\deg(v_1) + \deg(u_3) \leq 8$, there are at least eight edges between the vertices in V_2 and V_3 . Therefore, v_2 and v_3 are both adjacent to at least one of w_1 or w_2 . In this case, $\{u_1, w_1, w_2\}$ is a secure dominating set of H . In this subcase, therefore, $\gamma_s(H) = 3$ by (7.2).

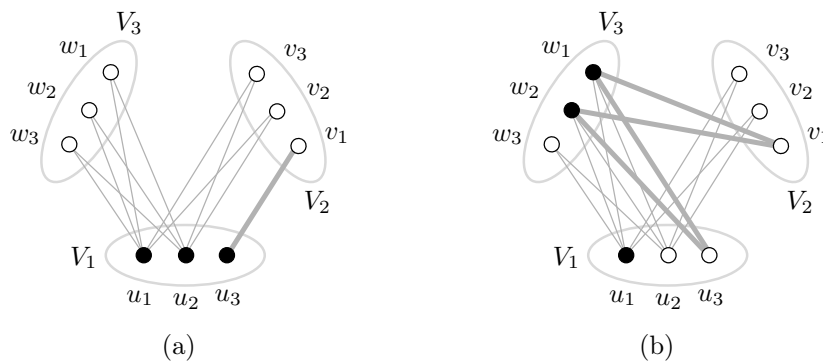


FIGURE 7.15: Subgraphs of H with $\Delta(H) = 5$ and $\delta(H) = 2$, where u_1 and u_2 have identical neighbourhoods and where (a) u_3 and v_1 are neighbours, with u_3 defending v_1 , and both u_1 and u_2 defending the vertices in $(V_2 - \{v_1\}) \cup V_3$, and (b) u_3 and v_1 are not neighbours, with u_1 defending w_3 , and w_1 and w_2 both defending $\{u_2, u_3, v_1, v_2, v_3\}$. A minimum secure dominating set is indicated by the solid vertices in each case.

Case ii(b): $N(u_1) \neq N(u_2)$. In this case u_1 is adjacent to a vertex, x (say), in $V_2 \cup V_3$ to which u_2 is not adjacent and, similarly, u_2 is adjacent to a vertex, y (say), in $V_2 \cup V_3$ to which u_1 is not adjacent.

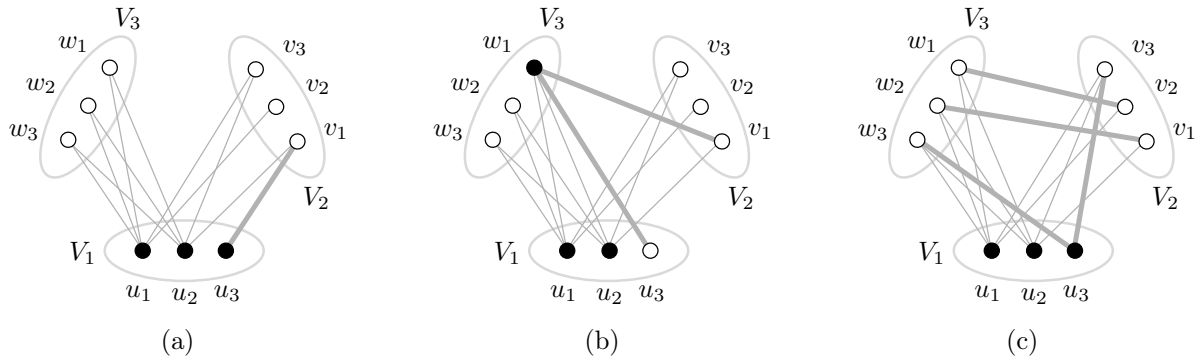


FIGURE 7.16: Subgraphs of H with $\Delta(H) = 5$ and $\delta(H) = 2$, where u_1 and u_2 do not have identical neighbourhoods and where (a) u_3 is adjacent to v_1 , with u_3 defending v_1 and u_2 defending $(V_2 - \{v_1\}) \cup V_3$, (b) u_3 is not adjacent to v_1 or v_2 , but shares w_1 as common neighbour with them, in which case w_1 defends u_3 , while u_2 defends $(V_2 - \{v_2, w_1\}) \cup V_3$ and u_2 defends v_2 , and (c) u_3 is not adjacent to v_1 or v_2 and shares no common neighbour with v_1 or v_2 , in which case u_3 defends $\{v_3, w_3\}$, u_2 defends $\{v_1, w_2\}$ and u_1 defends $\{v_2, w_1\}$. A minimum secure dominating set is indicated by the solid vertices in each case.

Suppose first that x and y are in the same partite set. It may be assumed, without loss of generality, that $x = v_1$ and $y = v_2$. Since $\delta(H) \geq 2$, u_3 is adjacent to at least two vertices in $V_2 \cup V_3$. If u_3 is adjacent to either v_1 or v_2 , then V_1 is a secure dominating set of H , as shown in Figure 7.16(a). If u_3 is adjacent to neither v_1 nor v_2 , but u_3 shares a neighbour, w_1 (say), with either v_1 or v_2 , then $\{u_1, u_2, w_1\}$ is a secure dominating set of H , as shown in Figure 7.16(b). Finally, if u_3 is adjacent to neither v_1 nor v_2 , and u_3 shares no common neighbours with v_1 and v_2 , then u_3, v_1 and v_2 are all adjacent to distinct vertices in V_3 and u_3 is also adjacent to w_3 . It may be assumed, without loss of generality, that v_1 is adjacent to w_2 and that v_2 is adjacent to w_1 . Then V_1 is a secure dominating set of H , as shown in Figure 7.16(c). In all of these cases, therefore, $\gamma_s(H) = 3$ by (7.2).

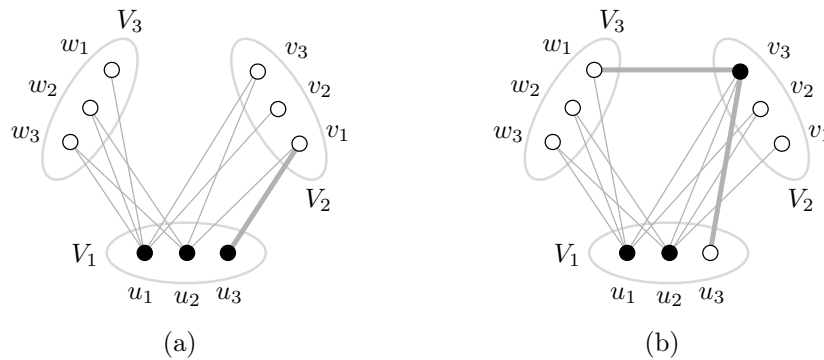


FIGURE 7.17: Subgraphs of H with $\Delta(H) = 5$ and $\delta(H) = 2$, where u_1 and u_2 do not have identical neighbourhoods and where (a) u_3 is adjacent to v_1 , in which case u_3 defends v_1 and u_2 defends $(V_2 - \{v_1, v_2\}) \cup V_3$, while u_1 defends v_2 , and (b) u_3 is adjacent to neither v_1 nor w_1 , but u_3 shares a neighbour v_3 with w_1 , in which case v_3 defends u_3 , u_2 defends v_1 and u_1 defends $(V_2 - \{v_1, v_3\}) \cup V_3$. A minimum secure dominating set is indicated by the solid vertices in each case.

Now suppose $x \in V_2$ and $y \in V_3$. It may be assumed, without loss of generality, that $x = v_1$ and $y = w_1$. If u_3 is adjacent to either v_1 or w_1 , or to both v_1 and w_1 , then V_1 is a secure dominating set of H , as shown in Figure 7.17(a). Suppose then that u_3 is adjacent to neither v_1 nor w_1 , but that u_3 shares a neighbour, z (say), with either v_1 or w_1 . It may be assumed,

without loss of generality, that $z = v_3$, in which case $\{u_1, u_2, v_3\}$ is a secure dominating set of H , as shown in Figure 7.17(b). If u_3 is adjacent to neither v_1 nor w_1 , and u_3 shares no common neighbours with v_1 and w_1 , but v_1 and w_1 are adjacent, then $\deg(v_1) + \deg(w_1) \leq 6$, in which case $\Delta(H) = 6$, a contradiction. Finally, suppose that u_3 is adjacent to neither v_1 nor w_1 , u_3 shares no common neighbours with v_1 and w_1 , and that v_1 and w_1 are not adjacent. It may be assumed, without loss of generality, that v_1 is adjacent to w_2 and that w_1 is adjacent to w_2 . Then $\deg(v_1) + \deg(w_1) = 4$, in which case $\Delta(H) = 6$, again a contradiction. In both of the cases that did not lead to contradictions, however, $\gamma_s(H) = 3$ by (7.2).

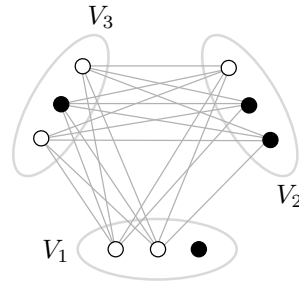


FIGURE 7.18: A graph $H \subseteq \mathcal{K}_{3,3,3} - 8e$ for which $\gamma_s(H) = 4$. A minimum secure dominating set is indicated by the solid vertices in each case.

From all of the above cases it is concluded that $\gamma_s(H) = 3$ and hence that $\mathcal{K}_{3,3,3}$ is p -stable for some $p \geq 7$. Finally, the subgraph of $\mathcal{K}_{3,3,3}$ in Figure 7.18 has size $27 - 8 = 19$ and secure domination number 4, showing that $\mathcal{K}_{3,3,3}$ is not stable for any $p \geq 8$. ■

The graph in Figure 7.19 is a member of $\mathcal{K}_{r,r,r} - (3r - 2)e$ which has no secure dominating set of cardinality 3 and hence $\mathcal{K}_{r,r,r} \notin \mathcal{S}_n^{n-2}$ for $n = 3r$, where $r \geq 3$. This shows that the result of Theorem 7.9 cannot be generalised to $\mathcal{K}_{r,r,r}$.

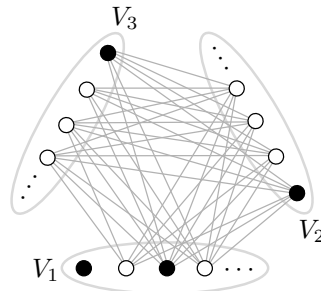


FIGURE 7.19: A member of $\mathcal{K}_{r,r,r} - (3r - 2)e$ which has no secure dominating set of cardinality 3. A minimum secure dominating set of cardinality 4 is indicated by the solid vertices.

The following result establishes a lower bound on ω_n .

Theorem 7.10 For any integer $n \geq 2$, $\mathcal{K}_{1,n-1} \in \mathcal{S}_n^{n-2}$.

Proof: The result is an immediate consequence of Theorem 6.12, which states that

$$C_p(\mathcal{K}_{1,n-1}) = \begin{cases} 0 & \text{if } 0 \leq p \leq n-2 \\ 1 & \text{if } p = n-1 \end{cases}$$

for the star graph $\mathcal{K}_{1,n-1}$ of order n . ■

The following result follows directly from Theorem 7.10.

Corollary 7.6 $\omega_n \geq n - 2$ for all $n \geq 2$.

There is another infinite class of $(n - 2)$ -stable graphs of order n , but in order to exhibit this class, the following result is required, which disqualifies large portions of any sufficiently large independent set of a graph from forming part of a minimum secure dominating set of the graph.

Theorem 7.11 *Let G be a graph for which $\gamma_s(G) = k$. If G contains an independent set S of cardinality at least $k + 1$, then every minimum secure dominating set of G contains at most $k - 2$ vertices of S .*

Proof: By contradiction. Suppose G contains an independent set S of cardinality at least $k + 1$ and let X be a minimum secure dominating set of G . There are two cases to consider:

Case i: $|X \cap S| = k - 1$. In this case X contains a single vertex, x (say), of $V(G) - S$ and there are at least two vertices, y and z (say), in the set $S - X$. Then x is adjacent to both y and z (for otherwise y and z are not dominated). However, since y and z are nonadjacent, Theorem 3.6 implies that they are not defended by x . But clearly y and z are also not defended by any vertex in $X - \{x\}$, a contradiction.

Case ii: $|X \cap S| = k$. In this case $X \subset S$. But then no vertex in $S - X$ is dominated, a contradiction.

These contradictions show that $|X \cap S| \leq k - 2$, as required. ■

It follows by the contrapositive of the special case where $k = 2$ in Theorem 7.11 that the secure domination number of the graph

$$H = \mathcal{K}_{1, \underbrace{3, 3, \dots, 3}_{\ell \text{ terms}}}$$

is at least 3. But since $H \in \mathcal{K}_n - 3\ell e = \mathcal{K}_n - (n - 1)e = (\mathcal{K}_n - e) - (n - 2)e$, for $n = 3\ell + 1$, it follows that $\mathcal{K}_n - e$ is not p -stable for any $p \geq n - 2$ if $n \equiv 1 \pmod{3}$. Therefore, $\mathcal{K}_n - e \notin \mathcal{S}_n^{n-2}$ if $n \equiv 1 \pmod{3}$.

Despite this infinite class of counter examples, it is demonstrated next that the graph class \mathcal{S}_n^{n-2} contains $\mathcal{K}_n - e$ if $n \not\equiv 1 \pmod{3}$.

Theorem 7.12

If $n \geq 3$ is a natural number such that $n \not\equiv 1 \pmod{3}$, then $\mathcal{K}_n - e \in \mathcal{S}_n^{n-2}$.

Proof: For any integer $n \geq 3$, the graph $\mathcal{K}_n - e$ contains a vertex v of degree $n - 1$. Removal of the $n - 1$ edges incident to v yields the graph $\mathcal{K}_1 \cup \mathcal{K}_{n-1} - e$. Since $\gamma_s(\mathcal{K}_n - e) = \gamma_s(\mathcal{K}_{n-1} - e) = 2$, but $\gamma_s(\mathcal{K}_1 \cup \mathcal{K}_{n-1} - e) = 1 + 2 = 3$, it therefore follows that

$$\mathcal{K}_n - e \text{ is not } p\text{-stable for any integer } p \geq n - 1. \quad (7.3)$$

Suppose now that $H \in \mathcal{K}_n - (n - 1)e$. Then it follows by the pigeonhole principle that at least two vertices of H have degree at least $n - 2$ each. There are two cases to consider:

Case i: At least one vertex of H has degree $n - 1$. There are three further subcases to consider in this case.

At least two vertices of H have degree $n - 1$ each. In this case any two universal vertices form a secure dominating set of H .

Exactly one vertex of H has degree $n - 1$ and at least one vertex of H has degree $n - 2$. In this case the universal vertex and any vertex of degree $n - 2$ form a secure dominating set of H .

Exactly one vertex of H , v (say), has degree $n - 1$, and no vertex of H has degree $n - 2$. In this case each vertex in $H - v$ has degree $n - 3$. Since every vertex of \overline{H} other than v therefore has degree 2, each component of $\overline{H} - v$ is a cycle. But since $n \not\equiv 1 \pmod{3}$, at least one of these cycles, C (say), is not a triangle. Let x, y and z be three consecutive vertices on C . Then x and z are adjacent in H and y is adjacent to neither x nor z in H . But since the degree of y in H is $n - 3$, y is therefore adjacent to all the vertices in $V(H) - \{x, y, z\}$. Hence the graph in Figure 7.20 is a subgraph of H and so $\{v, y\}$ is a secure dominating set of H , with v defending itself as well as x and z , and y defending itself and all the vertices in $V(H) - \{x, y, z, v\}$.

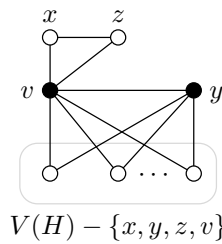


FIGURE 7.20: A subgraph of H for which x, y, z are three consecutive vertices on cycle of \overline{H} .

Case ii: No vertex of H has degree $n - 1$. In this case \overline{H} has size $n - 1$ and $\delta(\overline{H}) \geq 1$. Therefore at least one component of \overline{H} is a nontrivial tree T . Since all trees are bipartite, let V_1 and V_2 be the partite sets of T . Then the set $\{x, y\}$, where $x \in V_1$ and $y \in V_2$, is a secure dominating set of H , since x defends all the vertices in V_1 , y defends all the vertices in V_2 , and both x and y defend the vertices in $V(H) - V_1 - V_2$.

In all the above cases $\gamma_s(H) = 2$ and hence $\mathcal{K}_n - e$ is p -stable for some $p \geq n - 2$. The desired result therefore follows from (7.3). ■

It follows from Theorems 7.8–7.12 that the class of graphs

$$\Lambda_n = \begin{cases} \{\mathcal{K}_{1,7}, \mathcal{K}_8 - e, \mathcal{K}_{4,4}\} & \text{if } n = 8 \\ \{\mathcal{K}_{1,8}, \mathcal{K}_9 - e, \mathcal{K}_{3,3,3}\} & \text{if } n = 9 \\ \{\mathcal{K}_{1,n-1}\} & \text{if } n \equiv 1 \pmod{3} \\ \{\mathcal{K}_{1,n-1}, \mathcal{K}_n - e\} & \text{otherwise,} \end{cases} \quad (7.4)$$

is a subset of \mathcal{S}_n^{n-2} for all $n \geq 3$. Moreover, there is currently no evidence suggesting that there are any graphs in the class $\mathcal{S}_n^{n-2} \setminus \Lambda_n$, prompting the following conjecture.

Conjecture 7.3 $\mathcal{S}_n^{n-2} = \Lambda_n$ for all $n \geq 3$.

The final result of this section shows that the validity of Conjecture 7.3 would imply the validity of Conjecture 7.2.

Theorem 7.13 *If $\Lambda_n = \mathcal{S}_n^{n-2}$, then $\omega_n = n - 2$.*

Proof: By contradiction. Suppose $\Lambda_n = \mathcal{S}_n^{n-2}$, but assume, to the contrary that $\mathcal{S}_n^{n-1} \neq \emptyset$. Let $G \in \mathcal{S}_n^{n-1}$. Then it follows from Theorem 7.2 that there is a member $H \in G - 1e$ of the class \mathcal{S}_n^{n-2} such that $\gamma_s(G) = \gamma_s(H)$. There are three cases to consider:

Case i: $n \neq 8, 9$. In this case $H \not\cong \mathcal{K}_{1,n-1}$, since $\gamma_s(\mathcal{K}_{1,n-1}) = n - 1 > n - 2 = \gamma_s(\mathcal{K}_{1,n-1} + e)$ because of the triangle in $\mathcal{K}_{1,n-1} + e$. Furthermore, $H \not\cong \mathcal{K}_n - e$, since $\gamma_s(\mathcal{K}_n) = 1$ by Proposition 3.6, yet $\Phi(n - 1, 1, 0, 0)$ in Figure 4.7(a) is certificate showing that $\gamma_s(\mathcal{K}_n - e) = 2$. Therefore, there is a contradiction in this case.

Case ii: $n = 8$. In this case, additionally $H \not\cong \mathcal{K}_{4,4}$, because $\gamma_s(\mathcal{K}_{4,4}) = 4$ by Proposition 3.7, yet $\Psi(2, 1, 1, 4, 0, 0)$ is a certificate showing that $\gamma_s(\mathcal{K}_{4,4} + e) \leq 3$ by Theorem 4.3. This is again a contradiction.

Case iii: $n = 9$. In this case, additionally $H \not\cong \mathcal{K}_{3,3,3}$, because $\gamma_s(\mathcal{K}_{3,3,3}) = 3$ by Proposition 3.8, yet $\mathcal{K}_{3,3,3} + e$ is isomorphic to the graph in Figure 7.21, which admits $\{x, y\}$ as secure dominating set, showing that $\gamma_s(\mathcal{K}_{3,3,3} + e) \leq 2$, yet again a contradiction.

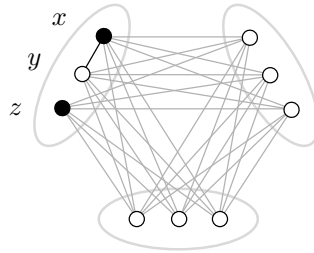


FIGURE 7.21: The graph $\mathcal{K}_{3,3,3} + e$, which admits $\{x, y\}$ as secure dominating set.

The above contradictions show that $\mathcal{S}_n^{n-1} = \emptyset$ and hence that $\omega_n \leq n - 2$. The desired result therefore follows from Corollary 7.6. ■

7.5 Special graph classes

In this section it is determined for which values of p and q members of a variety of special infinite graph classes are p -stable and q -critical. These special graph classes include complete graphs, complete bipartite graphs, paths, cycles, book graphs and dumbbells.

7.5.1 Complete and complete bipartite graphs

The first stability and criticality result follows directly from Theorem 4.1 and is an extension of Lemma 7.1.

Theorem 7.14 (Stability and criticality of complete graphs)

For any integer $n \geq 2$, the complete graph \mathcal{K}_n is 0-stable and 1-critical.

The next result is an extension of Theorem 7.10 and an immediate consequence of Theorem 6.12, which states that

$$c_q(\mathcal{K}_{1,n-1}) = C_q(\mathcal{K}_{1,n-1}) = \begin{cases} 0 & \text{if } 0 \leq q \leq n - 2 \\ 1 & \text{if } q = n - 1 \end{cases}$$

for the star graph $\mathcal{K}_{1,n-1}$ of order n .

Theorem 7.15 (Stability and criticality of stars)

For any integer $n \geq 2$, the star $\mathcal{K}_{1,n-1}$ is $(n-2)$ -stable and $(n-1)$ -critical.

Because removal of the spine edge from the book graph $\mathcal{B}_n = \mathcal{K}_2 + \overline{\mathcal{K}_{n-2}}$, shown in Figure 7.22, yields the complete bipartite graph $\mathcal{K}_{2,n-2}$, it is useful to establish the following result, which may be verified by examining the cost function values $c_2(\mathcal{B}_n) = 0$, $c_3(\mathcal{B}_n) = 1$, $C_1(\mathcal{B}_n) = 0$ and $C_2(\mathcal{B}_n) = 1$ motivated in Table 7.4.

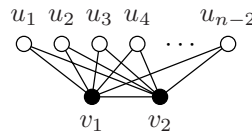


FIGURE 7.22: The labelled book graph $\mathcal{B}_n = \mathcal{K}_2 + \overline{\mathcal{K}_{n-2}}$.

Theorem 7.16 (Stability and criticality of book graphs)

For any integer $n \geq 5$, the book graph \mathcal{B}_n is 1-stable and 3-critical.

The next result follows immediately from Theorem 7.16 and Table 7.4.

Corollary 7.7 (Stability and criticality of $\mathcal{K}_{2,n-2}$)

For any integer $n \geq 5$, the complete bipartite graph $\mathcal{K}_{2,n-2}$ is 1-stable and 2-critical.

This section is closed by stating an immediate consequence of Corollary 7.7.

Corollary 7.8 (Stability and criticality of $\mathcal{K}_{2,n-2} - e$)

For any integer $n \geq 5$, the graph $\mathcal{K}_{2,n-2} - e$ is 0-stable and 1-critical.

7.5.2 Paths and cycles

In this subsection the stability and criticality values are determined for paths and cycles. The first result relates to the stability of paths and cycles.

Theorem 7.17 (Stability of paths and cycles)

- (a) *The only paths that are 1-stable are \mathcal{P}_3 and paths of the form $\mathcal{P}_{5+7\ell}$ for some $\ell \in \mathbb{N}_0$. All other paths are 0-stable.*
- (b) *The only cycles that are 2-stable are cycles of the form $\mathcal{C}_{5+7\ell}$ for some $\ell \in \mathbb{N}_0$. All other cycles, except \mathcal{C}_3 , are 1-stable. Finally, \mathcal{C}_3 is 0-stable.*

Proof: (a) It is easily verified exhaustively that the paths \mathcal{P}_2 and \mathcal{P}_4 are 0-stable, and that the paths \mathcal{P}_3 and \mathcal{P}_5 are 1-stable. Suppose, therefore, that $n \geq 6$. Then it follows from Proposition 3.9 that $\gamma_s(\mathcal{P}_n) = \lceil 3n/7 \rceil$. Removal of the fifth edge from \mathcal{P}_n yields the forest $\mathcal{P}_5 \cup \mathcal{P}_{n-5}$ for which

$$\gamma_s(\mathcal{P}_5 \cup \mathcal{P}_{n-5}) = \left\lceil \frac{3(5)}{7} \right\rceil + \left\lceil \frac{3(n-5)}{7} \right\rceil = 3 + \left\lceil \frac{3n}{7} - 2 - \frac{1}{7} \right\rceil = 1 + \left\lceil \frac{3n-1}{7} \right\rceil,$$

again by Proposition 3.9. Therefore $\gamma_s(\mathcal{P}_5 \cup \mathcal{P}_{n-5}) > \gamma_s(\mathcal{P}_n)$, unless $3n \equiv 1 \pmod{7}$, which has the unique solution $n \equiv 5 \pmod{7}$. For any $n \neq 7\ell + 5$, the path \mathcal{P}_n is therefore 0-stable.

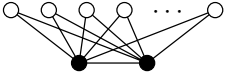
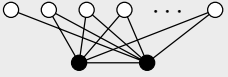

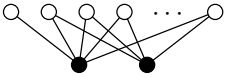
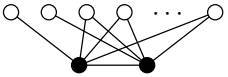
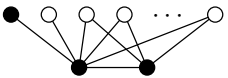
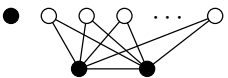


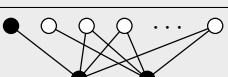



q	$\mathcal{B}_n - qe$	γ_s	$c_q(\mathcal{B}_n)$	$C_q(\mathcal{B}_n)$	Graphical representation
0	\mathcal{B}_n	2	0	0	
1	$\mathcal{B}_n - \{v_1u_1\}$	2	0	0	
	$\mathcal{B}_n - \{v_1v_2\}$	2			
2	$\mathcal{B}_n - \{v_1v_2, v_2u_1\}$	2	0	1	
	$\mathcal{B}_n - \{v_1u_2, v_2u_1\}$	3			
	$\mathcal{B}_n - \{v_2u_1, v_2u_2\}$	3			
	$\mathcal{B}_n - \{v_1u_1, v_2u_1\}$	3			
3	$\mathcal{B}_n - \{v_1v_2, v_1u_1, v_2u_1\}$	3	1	2	
	$\mathcal{B}_n - \{v_1v_2, v_2u_1, v_2u_2\}$	3			
	$\mathcal{B}_n - \{v_1v_2, v_1u_2, v_2u_1\}$	3			
	$\mathcal{B}_n - \{v_2u_1, v_2u_2, v_2u_3\}$	4			
	$\mathcal{B}_n - \{v_2u_1, v_2u_2, v_1u_3\}$	4			
	$\mathcal{B}_n - \{v_1u_1, v_2u_1, v_2u_2\}$	3			

TABLE 7.4: The cost function $c(\mathcal{B}_n) = 0, 0, 0, 1, \dots$ and $C(\mathcal{B}_n) = 0, 0, 1, 2, \dots$ for the book graph $\mathcal{B}_n = \mathcal{K}_2 + \overline{\mathcal{K}_{n-2}}$. Minimum secure dominating sets are denoted by solid vertices in each case. The same vertex labelling as in Figure 7.22 is adopted in the table.

If $n = 7\ell + 5$ for some $\ell \in \mathbb{N}$, then $\gamma_s(\mathcal{P}_n) = \lceil 3n/7 \rceil = 3\ell + 3$. Removal of the k -th edge from \mathcal{P}_n yields the forest $\mathcal{P}_k \cup \mathcal{P}_{n-k}$ for which

$$\gamma_s(\mathcal{P}_k \cup \mathcal{P}_{n-k}) = \left\lceil \frac{3k}{7} \right\rceil + \left\lceil \frac{3(7\ell+5-k)}{7} \right\rceil = \left\lceil \frac{3k}{7} \right\rceil + 3\ell + 2 + \left\lceil \frac{1-3k}{7} \right\rceil.$$

Since $\gamma_s(\mathcal{P}_n) \leq \gamma_s(\mathcal{P}_k \cup \mathcal{P}_{n-k})$, it follows that $\lceil 3k/7 \rceil + \lceil (1-3k)/7 \rceil \geq 1$. But by taking $a = 3k$ and $b = 7$ in the identity $\lceil a/b \rceil + \lceil (1-a)/b \rceil \leq 1$, which holds for any $a, b \in \mathbb{N}$ with $b \neq 0$ (see Proposition A.8 in Appendix A), it follows that $\lceil 3k/7 \rceil + \lceil (1-3k)/7 \rceil \leq 1$, so that, in fact, $\lceil 3k/7 \rceil + \lceil (1-3k)/7 \rceil = 1$. This implies that $\gamma_s(\mathcal{P}_n) = \gamma_s(\mathcal{P}_k \cup \mathcal{P}_{n-k})$ for any $k \in \{1, \dots, n-1\}$ if $n = 7\ell + 5$, showing that \mathcal{P}_n is p -stable for some $p \geq 1$ in this case.

However, removal of the first and second edges from $\mathcal{P}_{7\ell+5}$ yields the forest $2\mathcal{P}_1 \cup \mathcal{P}_{7\ell+3}$ for which $\gamma_s(2\mathcal{P}_1 \cup \mathcal{P}_{7\ell+3}) = 2\lceil 3/7 \rceil + \lceil 3(7\ell+3)/7 \rceil = 3\ell+4 > \gamma_s(\mathcal{P}_n)$, showing that $\mathcal{P}_{7\ell+5}$ is not p -stable for any $p > 1$. This therefore implies that $\mathcal{P}_{7\ell+5}$ is 1-stable.

(b) The stability results for cycles other than \mathcal{C}_3 follow immediately from (a) upon realisation that any edge removal from such a cycle produces a path of the same order. The cycle \mathcal{C}_3 is an exception merely because it is also a complete graph. ■

The criticality values of paths and cycles are established in the following result.

Theorem 7.18 (Criticality of paths and cycles) *Let n be a natural number.*

- (a) *Then the path \mathcal{P}_n is $(\frac{n}{2} - 5\lfloor \frac{n}{14} \rfloor)$ -critical if $n \geq 2$ is even, or $(\frac{n-5}{2} - 5\lfloor \frac{n-7}{14} \rfloor)$ -critical if $n \geq 7$ is odd. Finally, \mathcal{P}_3 is 2-critical and \mathcal{P}_5 is 3-critical.*
- (b) *Then the cycle \mathcal{C}_n is $(\frac{n+2}{2} - 5\lfloor \frac{n}{14} \rfloor)$ -critical if $n \geq 4$ is even, or $(\frac{n-3}{2} - 5\lfloor \frac{n-7}{14} \rfloor)$ -critical if $n \geq 7$ is odd. Finally, \mathcal{C}_3 is 1-critical and \mathcal{C}_5 is 3-critical.*

Proof: (a) Suppose $n \geq 2$ is even and let $q^* = \frac{n}{2} - 5\lfloor \frac{n}{14} \rfloor$. Then it follows from Theorem 6.8 that

$$C_q(\mathcal{P}_n) = \begin{cases} 0 & \text{if } q < \frac{n}{7} \\ \left\lceil \frac{2n+q+1}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil & \text{if } \frac{n}{7} \leq q \leq \frac{n}{2} \\ q+1 + \left\lceil \frac{3n}{7} \right\rceil & \text{if } q > \frac{n}{2}. \end{cases}$$

Note that

$$q^* = \frac{n}{2} - 5\left\lceil \frac{n-13}{14} \right\rceil \geq \left\lceil \frac{n}{7} + \frac{1}{2} \right\rceil + 3$$

by utilising the identities $\lfloor a/b \rfloor = \lceil (a-b+1)/b \rceil$ and $a - \lceil c \rceil \geq \lceil a-c \rceil - 1$ for any $a, b \in \mathbb{N}$ and any $c \in \mathbb{R}$ (see Corollary A.2 and Proposition A.6 in Appendix A). Therefore, $\frac{n}{7} \leq q^* - 1 < q^* \leq \frac{n}{2}$. It is next shown that $C_{q^*}(\mathcal{P}_n) = 1$, while $C_{q^*-1}(\mathcal{P}_n) = 0$. It follows, by the identity $\lfloor a/b \rfloor + \lceil c - (a/b) \rceil = c$ for any $a, b, c \in \mathbb{N}$ (see Proposition A.7 in Appendix A), that

$$\begin{aligned} C_{q^*}(\mathcal{P}_n) &= \left\lceil \frac{2n + \frac{n}{2} - 5\lfloor \frac{n}{14} \rfloor + 1}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \left\lfloor \frac{n}{14} \right\rfloor + \frac{1}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} + \frac{1}{5} \right\rceil - \left\lfloor \frac{n}{14} \right\rfloor - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} + \frac{1}{5} \right\rceil - \left\lfloor \frac{n}{14} \right\rfloor - \left\lceil \frac{7n}{14} - \frac{n}{14} \right\rceil \\ &= \left\lceil \frac{n}{2} + \frac{1}{5} \right\rceil - \frac{7n}{14} \\ &= \frac{n}{2} + 1 - \frac{n}{2}. \end{aligned}$$

Furthermore, it follows that

$$\begin{aligned} C_{q^*-1}(\mathcal{P}_n) &= \left\lceil \frac{2n + \frac{n}{2} - 5\lfloor \frac{n}{14} \rfloor}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \left\lfloor \frac{n}{14} \right\rfloor \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} \right\rceil - \left\lfloor \frac{n}{14} \right\rfloor - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} \right\rceil - \left\lfloor \frac{n}{14} \right\rfloor - \left\lceil \frac{7n}{14} - \frac{n}{14} \right\rceil \\ &= \left\lceil \frac{n}{2} \right\rceil - \frac{7n}{14} \\ &= \frac{n}{2} - \frac{n}{2}. \end{aligned}$$

(b) It is easily verified exhaustively that the paths \mathcal{P}_3 and \mathcal{P}_5 are 2-critical and 3-critical, respectively. Suppose therefore that $n \geq 7$ is odd and let $q' = \frac{n-5}{2} - 5\lfloor \frac{n-7}{14} \rfloor$. Note that

$$q' - 1 = \frac{n-5}{2} - 5\left\lceil \frac{n-20}{14} \right\rceil \geq \left\lceil \frac{n}{7} + \frac{1}{2} \right\rceil + 3$$

by utilising the identities $\lfloor a/b \rfloor = \lceil (a - b + 1)/b \rceil$ and $a - \lceil c \rceil \geq \lceil a - c \rceil - 1$ for any $a, b \in \mathbb{N}$ and any $c \in \mathbb{R}$ (see Corollary A.2 and Proposition A.6 in Appendix A). Therefore, $\frac{n}{7} \leq q' - 1 < q' \leq \frac{n}{2}$. It is next shown that $C_{q'}(\mathcal{P}_n) = 1$, while $C_{q'-1}(\mathcal{P}_n) = 0$. It follows, by the identity $\lfloor a/b \rfloor + \lceil c - (a/b) \rceil = c$ for any $a, b, c \in \mathbb{N}$ (see Proposition A.7 in Appendix A), that

$$\begin{aligned} C_{q'}(\mathcal{P}_n) &= \left\lceil \frac{2n + \frac{n-5}{2} - 5\lfloor \frac{n-7}{14} \rfloor + 1}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \left\lfloor \frac{n-7}{14} \right\rfloor - \frac{3}{10} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \frac{3}{10} \right\rceil - \left\lfloor \frac{n-7}{14} \right\rfloor - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \frac{3}{10} \right\rceil - \left\lfloor \frac{n-7}{14} \right\rfloor - \left\lceil \frac{7n-7}{14} - \frac{n-7}{14} \right\rceil \\ &= \left\lceil \frac{n}{2} - \frac{3}{10} \right\rceil - \frac{7n-7}{14} \\ &= \left\lceil \frac{n-1}{2} + \frac{1}{5} \right\rceil - \frac{n-1}{2} \\ &= \frac{n-1}{2} + 1 - \frac{n-1}{2}. \end{aligned}$$

Furthermore, it follows that

$$\begin{aligned} C_{q'-1}(\mathcal{P}_n) &= \left\lceil \frac{2n + \frac{n-5}{2} - 5\lfloor \frac{n-7}{14} \rfloor}{5} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \left\lfloor \frac{n-7}{14} \right\rfloor - \frac{1}{2} \right\rceil - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \frac{1}{2} \right\rceil - \left\lfloor \frac{n-7}{14} \right\rfloor - \left\lceil \frac{3n}{7} \right\rceil \\ &= \left\lceil \frac{n}{2} - \frac{1}{2} \right\rceil - \left\lfloor \frac{n-7}{14} \right\rfloor - \left\lceil \frac{7n-7}{14} - \frac{n-7}{14} \right\rceil \\ &= \left\lceil \frac{n}{2} - \frac{1}{2} \right\rceil - \frac{7n-7}{14} \\ &= \left\lceil \frac{n-1}{2} \right\rceil - \frac{n-1}{2} \\ &= \frac{n-1}{2} - \frac{n-1}{2}. \end{aligned}$$

(b) The criticality results for cycles other than \mathcal{C}_3 and \mathcal{C}_5 again follow from (a) upon realisation that any edge removal from such a cycle produces a path of the same order. ■

The results of Theorems 7.17 and 7.18 demonstrate that there exist connected p -stable, q -critical graphs for which the difference $q - p$ is arbitrarily large.

7.5.3 Dumbbells

The following result establishes the values of p and q for which the infinite graph class of dumbbells are p -stable and q -critical.

Theorem 7.19 *The dumbbell graph $\mathcal{D}_{i,n-i}$ is 0-stable and 2-critical for all $n \geq 4$ and all $2 \leq i \leq n-2$.*

Proof: Since the graph $\Phi(i-1, n-i, 1, 0)$ in Figure 4.7(d) is a spanning subgraph of $\mathcal{D}_{i,n-i}$, it follows from Theorem 4.2 that $\gamma_s(\mathcal{D}_{i,n-i}) = 2$. Furthermore, $\gamma_s(\mathcal{D}_{i,n-i} - e^*) = 2$, where e^* is the edge joining the vertex disjoint cliques of orders i and $n-i$, while $\gamma_s(\mathcal{D}_{i,n-i} - e) > 2$ for all $e \neq e^*$. Therefore, $C_0(\mathcal{D}_{i,n-i}) = 0$ and $C_1(\mathcal{D}_{i,n-i}) = 1$, so that $\mathcal{D}_{i,n-i}$ is 0-stable. Similarly, $c_1(\mathcal{D}_{i,n-i}) = 0$ and $c_2(\mathcal{D}_{i,n-i}) = 1$, so that $\mathcal{D}_{i,n-i}$ is 2-critical. ■

7.6 Chapter summary

The smallest number of edge removals from a graph that *necessarily* increases its secure domination number as well as the largest number of edge removals that does *not* increase its secure domination number, were explored in this chapter. The chapter opened in §7.1 with an introduction to the concepts of criticality and stability. Formal definitions of q -criticality and p -stability of a graph G were presented in terms of the two cost sequences $\mathbf{c}(G)$ and $\mathbf{C}(G)$ of Chapter 6. The notion of an edge-removal metagraph was also introduced, from which these cost sequences $\mathbf{c}(G)$ and $\mathbf{C}(G)$ may easily be determined for any graph G . The values of p and q for which a graph is p -stable and q -critical may easily be determined from its edge-removal metagraph. The section closed with the establishment of a result stating that both the criticality and stability values of a graph are non-increasing as one moves down in the edge-removal metagraph (as the values of p and q increase), provided that the secure domination number does not increase.

An inductive characterisation of the class \mathcal{Q}_n^q of q -critical graphs of order n was provided in terms of the class \mathcal{Q}_n^{q-1} in §7.2. This characterisation makes use of the 4-step construction procedure for the class \mathcal{Q}_n^1 by Grobler and Mynhard [55], as described in §3.2.3, as base case. An algorithm was then presented for iteratively computing the classes $\mathcal{Q}_n^2, \mathcal{Q}_n^3, \mathcal{Q}_n^4, \dots$ from \mathcal{Q}_n^1 . In a similar vein, a characterisation of the class \mathcal{S}_n^p of p -stable graphs of order n in terms of the class \mathcal{S}_n^{p-1} was provided in §7.3. While the problem of characterising the class \mathcal{S}_n^0 remains unsolved, a number of properties of 0-stable graphs were presented together with an algorithm for iteratively computing the classes $\mathcal{S}_n^1, \mathcal{S}_n^2, \mathcal{S}_n^3, \dots$ from the class \mathcal{S}_n^0 .

The focus in §7.4 shifted to establishing the largest values of p and q for which a graph of order n can be p -stable and q -critical (denoted by ω_n and Ω_n , respectively). In this section lower bounds on both Ω_n and ω_n were established, after which the values of these parameters were sought as functions of n . In each case such a function of n was conjectured. Furthermore, in the case of stability, a class Λ_n of graphs was presented which is conjectured to be the only class of graphs of order n that are Ω_n -stable.

The values of p and q for which a variety of infinite graph classes of special structure are p -stable and q -critical were then determined in §7.5. These special graph classes included complete graphs, complete bipartite graphs, paths, cycles, book graphs and dumbbells.

CHAPTER 8

Conclusion

Contents

8.1	Dissertation summary	165
8.2	Appraisal of dissertation contributions	167
8.3	Future work	168

The dissertation closes with a summary of the work contained therein, an appraisal of the contributions of the dissertation and a discussion on possibilities for related future work.

8.1 Dissertation summary

The dissertation opened in Chapter 1 with a brief description of the Dominating Queens Problem. An informal description of the notions of graph domination and secure graph domination was illustrated in the context of the celebrated Queen's graph, Q_8 . In the next section of Chapter 1, the notion of edge removal was considered and informal definitions of the notions of edge criticality and edge stability were provided with respect to edge removal in the context of secure graph domination. The chapter closed with a statement on the objectives to be pursued in the dissertation and a description of the organisation of material in the dissertation.

The basic mathematical concepts underlying the novel work in this dissertation were introduced in Chapter 2. This included the most basic fundamentals from graph theory in §2.1 and important properties relating to graph domination in §2.2. The chapter closed in §2.3 with a review of basic notions from complexity theory, which included recent algorithms for determining the domination number of an arbitrary graph.

A survey of the literature on topics related specifically to the protection of graphs was conducted in Chapter 3. Numerous graph protection strategies and their related parameters were reviewed in §3.1. A variety of results on secure graph domination were presented, which included a number of general bounds on the secure domination number of a graph, results on the exact values of the secure domination number for certain infinite classes of graphs and a number of bounds on the secure domination number for infinite classes of graphs for which the exact values of the secure domination number are yet unknown, in fulfilment of Dissertation Objective I of §1.3. The effect of edge removal was also considered in the context of secure graph domination, with a focus on criticality in secure graph domination. A number of variations on the notion of secure graph domination were reviewed in §3.3. More specifically, finite, higher-order generalisations were considered, which are applicable in the situation where more than a single attack occurs in

the graph. Generalisations were also considered for the cases where a vertex may contain more than one guard. The case where an infinite sequence of attacks occurs was considered as well, followed by generalisations allowing for the simultaneous movement of multiple guards during an attack.

In Chapter 4, a number of basic results on the nature and computation of minimum secure dominating sets of arbitrary graphs were established. The chapter opened with a description of three necessary and sufficient criteria for establishing whether or not a given subset of the vertex set of a graph is, in fact, a secure dominating set of the graph. Using these criteria, the classes of graphs that have secure domination numbers 1, 2 or 3 were characterised in fulfilment of Dissertation Objective II of §1.3. The class of graphs with secure domination number 1 was characterised as being the class of complete graphs only. For the class of graphs with secure domination number 2, a graph construction was required, resulting in a certificate denoted by $\Phi(i, j, k, \ell)$, as illustrated in Figure 4.2. It was shown that any incomplete graph which contains $\Phi(i, j, k, \ell)$ as spanning subgraph for some integers $i, j \geq 1$ and $k, \ell \geq 0$ has a secure domination number of 2. Certificates were presented showing that a number of well-known, infinite classes of graphs have secure domination number 2. A similar graph construction framework was presented for the class of graphs with secure domination number 3. In this case, however, the certificate denoted by $\Psi(i, j, k, r, s, t)$ is applicable for integers $i, j, k \geq 1$ and $r, s, t \geq 0$ as illustrated in Figures 4.8. Certificates were once again presented showing that a number of well-known, infinite classes of graphs admit secure dominating sets of cardinality 3.

Four algorithmic approaches for computing the secure domination number of a graph were put forward in Chapter 5, in fulfilment of Dissertation Objective III of §1.3. Two exact, exponential-time algorithms were presented for computing the secure domination number of an arbitrary graph. The first algorithm follows a branch-and-reduce approach, while the second algorithm adopts a branch-and-bound approach. The secure domination problem was also formulated as a binary programming problem. The binary programming formulation was solved using the software suite CPLEX 12.05 [37], while the other two algorithms mentioned above were implemented in Wolfram's Mathematica [106]. The execution times of the three above-mentioned algorithms were compared for different classes of small graphs. It was found that the binary programming solution approach is incomparably faster than the remaining two exact algorithms. Furthermore, of the remaining two algorithms the branch-and-bound algorithm outperformed the branch-and-reduce algorithm. A linear algorithm was finally presented for determining the secure domination number of an arbitrary tree. It was shown that this algorithmic approach may be implemented in linear space and time. This linear algorithm outperforms all three the above-mentioned algorithms when trees are considered.

The effects of multiple edge failures on the secure domination number of a graph were explored in Chapter 6. The chapter opened with a brief reference to practical applications of secure domination, as well as a result showing that edge removals from a graph cannot decrease the secure domination number of the resulting graph. Two cost functions, $c_q(G)$ and $C_q(G)$, of a graph G were introduced for measuring respectively the smallest possible and the largest possible increase in the secure domination number of a member of the set $G - qe$ over and above the value of $\gamma_s(G)$. The growth properties of these cost functions were analysed in Theorem 6.2. Some general bounds on the secure domination number were presented and used to derive lower bounds on $c_q(G)$ and upper bounds on $C_q(G)$ for any graph G of order n , thereby achieving Dissertation Objective IV of §1.3. The remainder of the chapter was devoted to establishing bounds on or values of the cost functions c_q and C_q for various infinite graph classes, in fulfilment of Dissertation Objective V of §1.3. These cost functions were determined exactly for paths and cycles. This was followed by establishing upper and lower bounds on $c_q(\mathcal{W}_n)$ and $C_q(\mathcal{W}_n)$ for a wheel \mathcal{W}_n .

of order n , respectively. The exact values of $c_q(\mathcal{K}_{1,n-1})$ and $C_q(\mathcal{K}_{1,n-1})$ were noted, after which the values of $c_q(\mathcal{K}_{2,n-2})$ and $C_q(\mathcal{K}_{2,n-2})$ were established. Good upper and lower bounds were also provided for $c_q(\mathcal{K}_{j,n-j})$ and $C_q(\mathcal{K}_{j,n-j})$, respectively, for $j > 2$. Finally, a combination of analytic and algorithmic bounds on the cost function $c_q(\mathcal{K}_n)$ and $C_q(\mathcal{K}_n)$ were presented.

In Chapter 7, the notions of criticality and stability were introduced, which measure respectively the smallest number of arbitrary edges whose deletion from a graph necessarily increases its secure domination number, and the largest number of arbitrary edges whose deletion necessarily does not increase its secure domination number. Formal definitions of the stability and criticality values associated with a graph G were presented in terms of the two cost functions $c_q(G)$ and $C_q(G)$ of Chapter 6. An inductive characterisation of the class \mathcal{Q}_n^q of q -critical graphs of order n was provided in terms of the class \mathcal{Q}_n^{q-1} and, in a similar vein, a characterisation of the class \mathcal{S}_n^p of p -stable graphs of order n in terms of the class \mathcal{S}_n^{p-1} was provided in fulfilment of Dissertation Objective VI of §1.3. Numerical results on the cardinalities of the nonempty graph classes of critical and stable graphs were presented for graph orders not exceeding 9. An investigation into establishing the largest values of p and q for which a graph of order n can be p -stable and q -critical (denoted by ω_n and Ω_n , respectively) was conducted. Lower bounds on both Ω_n and ω_n were established, after which the values of these parameters were sought as functions of n . In each case such a function of n was conjectured. Furthermore, in the case of stability, a class Λ_n of graphs was presented which is conjectured to be the only class of graphs of order n that are Ω_n -stable. The remainder of the chapter was devoted to determining the values of p and q for which a variety of infinite graph classes of special structure are p -stable and q -critical. These special graph classes included complete graphs, complete bipartite graphs, paths, cycles, book graphs and dumbbells.

8.2 Appraisal of dissertation contributions

The main contributions of this dissertation are fourfold. The first contribution centres around the characterisation of classes of graphs that have secure domination numbers 1, 2 or 3. This was possible due to the result of Theorem 3.6, which was used to partition the vertex set of a graph G into five subsets with respect to any subset X of the vertex set of G . This work has been submitted for publication [21].

The second contribution was the design and subsequent analysis of various exact algorithmic approaches towards computing the secure domination number of an arbitrary graph. The designs of a branch-and-reduce algorithm and a branch-and-bound algorithm were inspired by the work of Van Rooij and Bodlaender [97] in the context of classical graph domination, who based their algorithm on the minimum set cover problem and used a series of reduction rules as a design tool to refine their approach. This work has been published in [15]. Furthermore, a novel binary programming model formulation was presented for determining the secure domination number of a graph. This work has been published in [14].

The algorithmic contributions of this dissertation also include a linear algorithm for secure domination of trees, which was inspired by the work of Cockayne, Goodman and Hedetniemi for the classical domination of trees [31]. The algorithmic approach towards finding a minimum secure dominating set of an arbitrary tree T entails including the vertices required in a minimum secure dominating set of a pendent spider \mathcal{S} of T , pruning away \mathcal{S} from T to form a smaller tree T' and repeating this process for T' until only a final spider remains. It was shown that this algorithm may be implemented in both linear space and time. This work has been published in [17].

The third contribution involves establishing results on the two novel cost functions, c_q and C_q ,

mentioned above. It was possible to establish general bounds on the secure domination number of a graph which, in turn, was used to derive general bounds on these cost functions. The focus then shifted to determining exact values for or tight bounds on the cost functions c_q and C_q for various infinite graph classes. This work has been accepted for publication in [16].

The final contribution involves the introduction of the notions of criticality and stability with respect to edge-removal in the context of secure graph domination. The notion of an edge-removal metagraph of a complete graph of order n presented a natural, albeit computationally expensive, framework for establishing the nonempty classes of critical and stable graphs of order n . An inductive characterisation of q -critical graphs was established and this characterisation was used to derive a computationally cheaper algorithm for computing all q -critical graphs of small order. A similar result was possible for stability, although the problem of characterising 0-stable graphs of order n remains open. The results of the algorithmic implementations for computing the nonempty classes of critical and stable graphs of order n were compared to the results obtained by means of the edge-removal metagraph \mathcal{K}_n in order to validate the numerical results. The empirical establishment of extremal stability and criticality values led to lower bounds on the parameters Ω_n and ω_n of §7.2 and §7.3, respectively, for small values of n , after which the values of these parameters were sought as functions of n in general. In each case an appropriate function of n was conjectured. This contribution finally entailed determining the values of p and q for which members of a variety of special infinite graph classes are p -stable and q -critical. The work on the notion of criticality has been accepted for publication in [18, 19], while work on the notion of stability has also been submitted for publication in [20].

8.3 Future work

In this section, four open questions related to the secure domination number of a graph are posed. Seven suggestions are also made with respect to possible future research emanating from the work presented in this dissertation.

A characterisation of secure dominating sets by Cockayne *et al.* [32] was reviewed in Theorem 3.6. This characterisation was then used in §4.1 to devise three necessary and sufficient criteria for establishing whether or not a subset of the vertex set of a graph G is, in fact, a secure dominating set of G . These criteria were used to partition the vertex set of G into five subsets with respect to any subset X of the vertex set of G , as demonstrated in Figure 4.1. This was instrumental in characterising the classes of graphs with secure domination numbers 1, 2 and 3. These three criteria were once again used in the branch-and-reduce algorithm of §5.1.1 and the branch-and-bound algorithm of §5.1.2 for computing the secure domination number of an arbitrary graph. Theorem 3.6, therefore, underpins many of the contributions presented in this dissertation on secure graph domination.

Question 8.1 *Is it possible to adapt the characterisation of secure dominating sets in Theorem 3.6 so as to be applicable to other domination parameters in which swap sets are required to be dominated, such as weak Roman domination or k -secure domination?*

The structure of a secure dominating set, as described in §4.1, was instrumental in characterising graphs with small secure domination numbers.

Question 8.2 *Is it possible to establish characterisations for graphs with secure domination number 4 or 5 that are simple enough to use in a practical way?*

Both the branch-and-reduce algorithm of §5.1.1 and the the branch-and-bound algorithm of §5.1.2 use the two rules in Theorem 5.1, which exclude certain vertices when seeking a minimum secure dominating set of a graph. These two reduction rules were inspired by the reduction rules of Grandoni [54, Lemma 1] for the celebrated set cover problem.

Question 8.3 *Is it possible to improve on the current reduction rules used in Algorithms 5.2 and 5.4 by generalising the reduction rules for graph domination by Van Rooij and Bodlaender [97] to reduction rules for secure graph domination? How will the use of such improved or additional reduction rules, in conjunction with low-level computer implementations of Algorithms 5.2 and 5.4, compare to the efficiency of the binary programming implementation in §5.1.5?*

To the author's best knowledge, Algorithm 5.6 is the first fully linear algorithm for determining a domination parameter for a tree T in which a swap set is required to be a dominating set of T .

Question 8.4 *Is it possible to use the results in §5.2.3 to design a linear algorithm for determining the weak Roman domination number of an arbitrary tree?*

The first pertinent suggestion for future research is to determine the value of Ω_n for all $n \geq 10$. The result of Theorem 7.5 and the circumstantial numerical evidence in Table 7.2 suggest the result of Conjecture 7.1.

Suggestion 8.1 *Prove or disprove Conjecture 7.1. In view of Observation 7.1 and Theorem 7.5, one would have to show that there exists no graph of order n that is $\left(\binom{n}{2} - 2n + 6\right)$ -critical in order to prove the conjecture correct, or produce a graph that is q -critical for some $q \in \left\{\binom{n}{2} - 2n + 6, \dots, \binom{n}{2}\right\}$ in order to refute the conjecture.*

It was shown in §7.3 how the graph classes $\mathcal{S}_n^1, \dots, \mathcal{S}_n^{\omega_n}$ may be computed inductively from the class \mathcal{S}_n^0 for all $n \geq 2$.

Suggestion 8.2 *Design of a stepwise construction process for the graph class \mathcal{S}_n^0 .*

The graph class Λ_n in (7.4) is a subclass of \mathcal{S}_n^{n-2} for all $n \geq 2$. The author believes that $\mathcal{S}_n^{n-2} = \Lambda_n$ for all $n \geq 2$ (Conjecture 7.3), but is unable to prove this.

Suggestion 8.3 *Prove Conjecture 7.3 or demonstrate a graph in the class $\mathcal{S}_n^{n-2} \setminus \Lambda_n$.*

Although the proof of Conjecture 7.3 would establish the truth of Conjecture 7.2 (by Theorem 7.13), it may of course be possible that Conjecture 7.2 is true even if Conjecture 7.3 is false.

Suggestion 8.4 *Resolve the truth or otherwise of Conjecture 7.2 as an independent problem.*

In terms of the cost functions of Chapter 6, a graph G is q -critical if $c_q(G) > 0$, but $c_{q-1}(G) = 0$, while G is p -stable if $C_p(G) = 0$, but $C_{p+1}(G) > 0$. From a practical perspective it may, however, be useful to investigate the value of q for which $c_q(G) > k$, but $c_{q-1}(G) = k$, as well as the value of p for which $C_p(G) = \ell$, but $C_{p+1}(G) > \ell$, for all $k, \ell \in \{0, \dots, n - \gamma_s(G) - 1\}$. The

following generalised definitions of the notions of criticality and stability may be adopted in this case. A graph G is (q, k) -critical if the *smallest* arbitrary subset of edges whose removal from G necessarily increases the secure domination number by at least k units, has cardinality q . A graph G is (p, ℓ) -stable if the *largest* subset of arbitrary edges whose removal from G necessarily does *not* increase the secure domination number of the resulting graph by at least ℓ units, has cardinality p .

Suggestion 8.5 Compute the classes of (q, k) -critical graphs of order n for all admissible values of q and k , as well as the classes of (p, ℓ) -stable graphs of order n for all admissible values of p and ℓ .

From a practical perspective, it may also be helpful to quantify the benefit (in terms of the number of edge failures that may be accommodated) when investing in a specified number of guards in addition to the minimum number required for the secure domination of a graph G . Let $d_i(G)$ be the largest number of edge removals from G that may potentially be accommodated when securely dominating G with $i \in \mathbb{N}_0$ additional guards over and above the minimum number $\gamma_s(G)$. That is, $d_i(G)$ is the largest value of q for which there exists a graph $H \in G - qe$ such that $\gamma_s(H) = \gamma_s(G) + i$.

In a similar vein, let $D_i(G)$ be the largest number of edge removals from the graph G that can necessarily be accommodated when securely dominating G with $i \in \mathbb{N}_0$ additional guards over and above the minimum number $\gamma_s(G)$. Then $D_i(G)$ is the largest value of q such that $\gamma_s(H) \leq \gamma_s(G) + i$ for all $H \in G - qe$.

The sequences

$$\mathbf{d}(G) = d_0(G), d_1(G), \dots, d_{n-\gamma_s(G)}(G) \quad \text{and} \quad \mathbf{D}(G) = D_0(G), D_1(G), \dots, D_{n-\gamma_s(G)}(G)$$

of benefit functions may each be thought of as step functions (with steps of unit height but variable length). For example, it may be seen from Table 6.1 that $\mathbf{d}(\mathcal{P}_6) = 2, 3, 4, 5$ and $\mathbf{D}(\mathcal{P}_6) = 0, 2, 4, 5$ for the path \mathcal{P}_6 of order 6.

Suggestion 8.6 Investigate the benefit sequences $\mathbf{d}(G)$ and $\mathbf{D}(G)$, establishing general bounds on these sequences or exact values of the sequences for special infinite graph classes.

In applications conforming to the notion of secure domination, one may seek the benefit (in terms of the reduction in the number of guards with respect to $\gamma_s(G)$) if a number of edges were to be added between non-adjacent vertices of a graph G . In this case, the two alternative benefit functions

$$\begin{aligned} \text{and} \quad b_q(G) &= \gamma_s(G) - \max \gamma_s(G + qe) \\ B_q(G) &= \gamma_s(G) - \min \gamma_s(G + qe) \end{aligned}$$

are non-negative in view of Corollary 6.1 and measure respectively the *smallest possible* and the *largest possible* decrease in the minimum number of guards required to dominate a member of $G + qe$ securely, in the event that an arbitrary set of $0 \leq q \leq \binom{n}{2} - m$ edges were to be added between non-adjacent vertices of G . The sequences

$$\mathbf{b}(G) = b_0(G), b_1(G), \dots, b_{\binom{n}{2}-m}(G) \quad \text{and} \quad \mathbf{B}(G) = B_0(G), B_1(G), \dots, B_{\binom{n}{2}-m}(G)$$

of benefit functions may each be thought of as step functions (with steps of unit size) for any graph G of size m .

An *edge-alteration metagraph* may be associated with a graph G of size m . The nodes of this metagraph represent the non-isomorphic members of $G - qe$ for all $q = 0, 1, \dots, m$ and $G + qe$ for all $q = 0, 1, \dots, \binom{n}{2} - m$, and may be arranged in $\binom{n}{2} + 1$ levels. The benefit sequences $\mathbf{b}(G)$ and $\mathbf{B}(G)$ can easily be determined from the edge-alteration metagraph of G . The edge-alteration metagraph of the path \mathcal{P}_4 of order 4 is shown as an example in Figure 8.1.

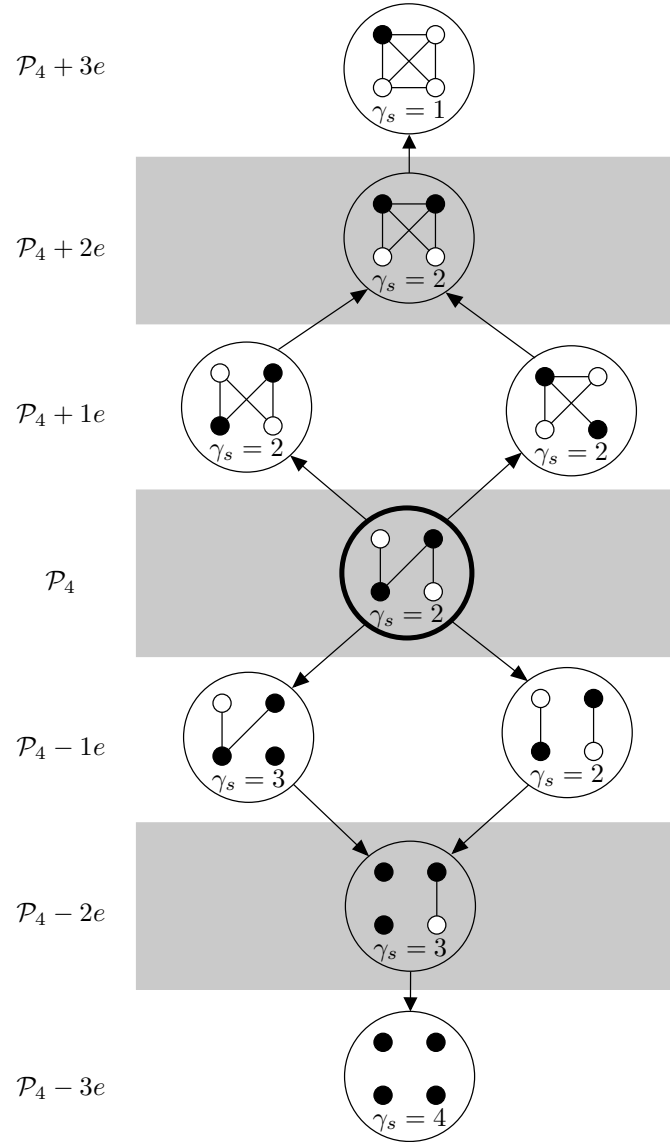


FIGURE 8.1: The edge-alteration metagraph of the path graph \mathcal{P}_4 of order 4. It follows that the sequences of benefit functions of \mathcal{P}_4 are $\mathbf{b}(\mathcal{P}_4) = \mathbf{B}(\mathcal{P}_4) = 0, 0, 0, 1$.

Suggestion 8.7 Study the notion of edge addition with respect to secure graph domination. Investigate the benefit sequences $\mathbf{b}(G)$ and $\mathbf{B}(G)$, establishing general bounds on these sequences or exact values of the sequences for special infinite graph classes.

References

- [1] ALON N & SPENCER JH, 1992, *The probabilistic method*, John Wiley & Sons, Hoboken (NJ).
- [2] ARNAUTOV VI, 1974, *Estimation of the exterior stability number of a graph by means of the minimum degree of the vertices* (in Russian), *Prikladnaya Matematika i Programirovanie*, **11**, pp. 3–8.
- [3] BAUER D, HARARY F, NIEMINEN J & SUFFEL CL, 1983, *Domination alteration sets in graphs*, *Discrete Mathematics*, **47**, pp. 153–161.
- [4] BENECKE S, 2004, *Higher order domination of graphs*, MSc Thesis, Stellenbosch University, Stellenbosch.
- [5] BENECKE S, GROBLER PJP & VAN VUUREN JH, 2007, *Secure total domination in graphs*, *Utilitas Mathematica*, **74**, pp. 247–259.
- [6] BERGE C, 1962, *Theory of graphs and its applications*, Methuen, London.
- [7] BOLLOBÁS B & COCKAYNE EJ, 1979, *Graph-theoretic parameters concerning domination, independence, and irredundance*, *Journal of Graph Theory*, **3**(3), pp. 241–249.
- [8] BONDY JA & MURTY USR, 2008, *Graduate texts in mathematics: Graph theory*, Springer, New York (NY).
- [9] BRIGHAM RC, CHINN PZ & DUTTON RP, 1988, *Vertex domination-critical graphs*, *Networks*, **18**, pp. 173–179.
- [10] BRIGHAM RC & DUTTON RP, 1990, *Bounds on the domination number of a graph*, *The Quarterly Journal of Mathematics*, **41**(3), pp. 269–275.
- [11] BROOKS SL, 1941, *On colouring the nodes of a network*, *Cambridge Philosophical Society*, **37**, pp. 194–197.
- [12] BURGER AP, COCKAYNE EJ, GRÜNDLINGH WR, MYNHARDT CM, VAN VUUREN JH & WINTERBACH W, 2004, *Finite order domination in graphs*, *Journal of Combinatorial Mathematics and Combinatorial Computing*, **49**, pp. 159–175.
- [13] BURGER AP, COCKAYNE EJ, GRÜNDLINGH WR, MYNHARDT CM, VAN VUUREN JH & WINTERBACH W, 2004, *Infinite order domination in graphs*, *Journal of Combinatorial Mathematics and Combinatorial Computing*, **50**, pp. 179–194.
- [14] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, 2013, *A binary programming approach towards achieving effective graph protection*, *Proceedings of the 42nd Annual Conference of the Operations Research Society of South Africa*, Stellenbosch, pp. 19–30.
- [15] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, 2013, *Two algorithms for secure graph domination*, *Journal of Combinatorial Mathematics and Combinatorial Computing*, **85**, pp. 321–339.

- [16] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, 2014, *The cost of edge failure with respect to secure domination*, Utilitas Mathematica, To appear.
- [17] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, 2014, *A linear algorithm for secure domination in trees*, Discrete Applied Mathematics, **171**, pp. 15–27.
- [18] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, 2014, *On the q -criticality of graphs with respect to secure graph domination*, Proceedings of the 43rd Annual Conference of the Operations Research Society of South Africa, Parys, To appear.
- [19] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, *Edge criticality in secure graph domination*, Submitted.
- [20] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, *Edge stability in secure graph domination*, Submitted.
- [21] BURGER AP, DE VILLIERS AP & VAN VUUREN JH, *On minimum secure dominating sets of graphs*, Submitted.
- [22] BURGER AP, HENNING MA & VAN VUUREN JH, 2009, *Vertex covers and secure domination in graphs*, Quaestiones Mathematicae, **31(2)**, pp. 163–171.
- [23] BURROUGH PA & McDONNELL RA, 1998, *Principles of geographical information systems*, Oxford University Press, Oxford.
- [24] CAPRARA A, TOTH P & FISCHETTI M, 2000, *Algorithms for the set covering problem*, Annals of Operations Research, **98**, pp. 353–371.
- [25] CHARTRAND G & OELLERMANN O, 1993, *Applied and algorithmic graph theory*, McGraw-Hill, New York (NY).
- [26] CHVÁTAL V & COOK W, 1990, *The discipline of a graph*, Discrete Mathematics, **86**, pp. 191–198.
- [27] COCKAYNE EJ, 2007, *Irredundance, secure domination and maximum degree in trees*, Discrete Mathematics, **307**, pp. 12–17.
- [28] COCKAYNE EJ, DAWES RM & HEDETNIEMI ST, 1980, *Total domination in graphs*, Networks, **10**, pp. 211–219.
- [29] COCKAYNE EJ, DREYER PA, HEDETNIEMI SM & HEDETNIEMI ST, 2004, *Roman domination in graphs*, Discrete Mathematics, **278(1–3)**, pp. 11–22.
- [30] COCKAYNE EJ, FAVARON O & MYNHARDT CM, 2003, *Secure domination, weak Roman domination and forbidden subgraphs*, Bulletin of the Institute of Combinatorics and its Applications, **39**, pp. 87–100.
- [31] COCKAYNE EJ, GOODMAN S & HEDETNIEMI S, 1975, *A linear algorithm for the domination number of a tree*, Information Processing Letters, **4(2)**, pp. 41–44.
- [32] COCKAYNE EJ, GROBLER PJP, GRÜNDLINGH WR, MUNGANGA J & VAN VUUREN JH, 2005, *Protection of a graph*, Utilitas Mathematica, **67**, pp. 19–32.
- [33] COCKAYNE EJ & HEDETNIEMI S, 1977, *Towards the theory of domination in graphs*, Networks, **7**, pp. 247–261.
- [34] COCKAYNE EJ & HEDETNIEMI ST, 1974, *Independence graphs*, Proceedings of the 5th Southeast Conference on Combinatorics, Graph Theory and Computing, Winnipeg, pp. 471–491.
- [35] COCKAYNE EJ, HEDETNIEMI ST & MILLER DJ, 1978, *Properties of hereditary hypergraphs and middle graphs*, Canadian Mathematical Bulletin, **24(1)**, pp. 461–468.

- [36] COOK SA, 1971, *The complexity of theorem-proving procedures*, Proceedings of the 3rd ACM Symposium on Theory of Computing, Shaker Heights (OH), pp. 151–158.
- [37] CPLEX EDITION: VERSION 12.5, 2012, *IBM ILOG CPLEX Optimization Studio*, [Online], [Cited April 15th, 2013], Available from <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>.
- [38] CREIGNOU N, KHANNA S & SUDAN M, 2001, *Complexity classification of boolean constraint satisfaction problems*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (PA).
- [39] DE JAENISCH CF, 1862, *Applications de l'Analyse Mathematique au Jeu des Echecs* (in French), Petrograd.
- [40] EDMONDS J, 1965, *Paths, trees, and flowers*, Canadian Journal of Mathematics, **17**, pp. 449–467.
- [41] FAVARON O, HENNING MA, MYNHART CM & PUECH J, 2000, *Total domination in graphs with minimum degree three*, Journal of Graph Theory, **34(1)**, pp. 9–19.
- [42] FAVARON O, SUMNER D & WOJCICKA E, 1994, *The diameter of domination k-critical graphs*, Journal of Graph Theory, **18(7)**, pp. 723–734.
- [43] FINK JF, JACOBSON MS, KINCH LF & ROBERTS J, 1985, *On graphs having domination number half their order*, Periodica Mathematica Hungarica, **16(4)**, pp. 287–293.
- [44] FINK JF, JACOBSEN MS, KINCH LF & ROBERTS J, 1990, *The bondage number of a graph*, Discrete Mathematics, **86**, pp. 47–57.
- [45] FLACH P & VOLKMANN L, 1990, *Estimations for the domination number of a graph*, Discrete Mathematics, **80(2)**, pp. 145–151.
- [46] FOMIN FV, KRATSCH D & WOEGINGER JG, 2004, *Exact (exponential) algorithms for the dominating set problem*, Proceedings of the 30th Workshop on Graph Theoretic Concepts in Computer Science, Bad Honnef, pp. 245–256.
- [47] FOMIN FV, GRANDONI F & KRATSCH D, 2005, *Some new techniques in design and analysis of exact (exponential) algorithms*, Bulletin of the European Association for Theoretical Computer Science, **87**, pp. 47–77.
- [48] FOULDS LR, 1992, *Graph theory applications*, Springer-Verlag, New York (NY).
- [49] GAREY MR & JOHNSON DS, 1979, *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, New York (NY).
- [50] GIBBONS A, 1985, *Algorithmic graph theory*, Cambridge University Press, Cambridge.
- [51] GODDARD W, HEDETNIEMI SM & HEDETNIEMI ST, 2005, *Eternal security in graphs*, Journal of Combinatorial Mathematics and Combinatorial Computing, **52**, pp. 169–180.
- [52] GOLDWASSER JL & KLOSTERMEYER WF, 2008, *Tight bounds for eternal dominating sets in graphs*, Discrete Mathematics, **308**, pp. 2589–2593.
- [53] GRAHAM RL, KNUTH DE & PATASHNIK O, 1989, *Concrete mathematics*, Addison-Wesley, New York (NY).
- [54] GRANDONI F, 2006, *A note on the complexity of minimum dominating set*, Journal of Discrete Algorithms, **4(2)**, pp. 209–214.
- [55] GROBLER PJP & MYNHARDT CM, 2009, *Secure domination critical graphs*, Discrete Mathematics, **309**, pp. 5820–5827.
- [56] HARANT J, PRUCHNEWSKI A & VOIGT M, 1999, *On dominating sets and independent sets in graphs*, Combinatorics, Probability and Computing, **8(6)**, pp. 547–553.

- [57] HARARY F & HAYNES T, 2000, *Double domination in graphs*, Ars Combinatoria, **55**, pp. 201–213.
- [58] HARTNELL BL & RALL DF, 1994, *Bounds on the bondage number of a graph*, Discrete Mathematics, **128**, pp. 173–177.
- [59] HAYNES TW, HEDETNIEMI ST & SLATER PJ, 1998, *Domination in graphs: Advanced topics*, Marcel Dekker, New York (NY).
- [60] HAYNES TW, HEDETNIEMI ST & SLATER PJ, 1998, *Fundamentals of domination in graphs*, Marcel Dekker, New York (NY).
- [61] HENNING MA, 2009, *A survey of selected recent results on total domination in graphs*, Discrete Mathematics, **309**(1), pp. 32–63.
- [62] HENNING MA & HEDETNIEMI ST, 2003, *Defending the Roman Empire — A new strategy*, Discrete Mathematics, **266**, pp. 239–251.
- [63] HENNING MA & VAN VUUREN JH, 2013, *Graph and network theory: An applied approach using mathematica*, In Preparation, Stellenbosch.
- [64] HENNING MA & YEO A, 2013, *Total domination in graphs*, Springer Science and Business Media, New York (NY).
- [65] KEIL JM, 1986, *Total domination in interval graphs*, Information Processing Letters, **22**, pp. 171–174.
- [66] KLOSTERMEYER WF & MACGILLIVRAY G, 2007, *Eternal security in graphs of fixed independence number*, Journal of Combinatorial Mathematics and Combinatorial Computing, **63**, pp. 97–101.
- [67] KLOSTERMEYER WF & MACGILLIVRAY G, 2009, *Eternal dominating sets in graphs*, Journal of Combinatorial Mathematics and Combinatorial Computing, **68**, pp. 97–111.
- [68] KLOSTERMEYER WF & MYNHARDT CM, 2008, *Secure domination and secure total domination in graphs*, Discussiones Mathematicae, **28**, pp. 267–284.
- [69] KLOSTERMEYER WF & MYNHARDT CM, 2011, *Graphs with equal vertex cover and eternal domination numbers*, Discrete Mathematics, **311**, pp. 1371–1379.
- [70] KLOSTERMEYER WF & MYNHARDT CM, 2012, *Eternal total domination in graphs*, Ars Combinatoria, **107**, pp. 473–492.
- [71] KLOSTERMEYER WF & MYNHARDT CM, 2012, *Vertex covers and eternal domination*, Discrete Applied Mathematics, **160**, pp. 1183–1190.
- [72] KOK J & MYNHARDT CM, 1990, *Reinforcement in graphs*, Congressus Numerantium, **79**, pp. 225–231.
- [73] LASKAR R, PFAFF J, HEDETNIEMI SM & HEDETNIEMI ST, 1984, *On the algorithmic complexity of total domination*, SIAM Journal on Algebraic and Discrete Methods, **5**(3), pp. 420–425.
- [74] LUTTWAK EN, 1976, *The grand strategy of the Roman Empire*, Johns Hopkins University Press, Baltimore (MD).
- [75] MACGILLIVRAY G & SEYFFARTH K, 1996, *Domination numbers of planar graphs*, Journal of Graph Theory, **22**(3), pp. 213–229.
- [76] MCCUAIG W & SHEPHERD B, 1989, *Domination in graphs with minimum degree two*, Journal of Graph Theory, **13**(6), pp. 749–762.
- [77] MITCHELL K, 2007, *Bounds on the chromatic number*, MSc Thesis, Colorado University, Denver (CO).

- [78] MITCHELL SL, COCKAYNE EJ & HEDETNIEMI S, 1979, *Linear algorithms on recursive representations of trees*, Journal of Computer and System Sciences, **18(1)**, pp. 76–85.
- [79] MYNHARDT CM, SWART HC & UNGERER E, 2005, *Excellent trees and secure domination*, Utilitas Mathematica, **67**, pp. 255–267.
- [80] ORE O, 1962, *Theory of graphs*, American Mathematical Society, Colloquium Publications, Providence (RI).
- [81] PAPADIMITRIOU CH, 1995, *Computational complexity*, Addison-Wesley Publishing Company, Reading (MA).
- [82] PAYAN C, 1975, *Sur le nombre d'absorption d'un graphe simple* (in French), Cahiers du Centre d'Études de Recherche Opérationnelle, **2.3.4**, pp. 171.
- [83] PAYAN C & XUONG NH, 1982, *Domination-balanced graphs*, Journal of Graph Theory, **6(1)**, pp. 23–32.
- [84] PERLA PP, 1990, *The art of wargaming*, Naval Institute Press, Annapolis (MD).
- [85] PETERSON I, 2000, *Math trek: Defending the Roman Empire*, [Online], [Cited February 9th, 2013], Available from <http://www.sciencenews.org/view/download/id/26311/name/.jpg>.
- [86] REED B, 1996, *Paths, stars and the number three*, Combinatorics, Probability and Computing, **5(3)**, pp. 277–295.
- [87] REVELLE CS & ROSING KE, 2000, *Defendens imperium Romanum: A classical problem in military strategy*, The American Mathematical Monthly, **107(7)**, pp. 585–594.
- [88] SCHIERMEYER I, 2008, *Efficiency in exponential time for domination-type problems*, Discrete Applied Mathematics, **156(17)**, pp. 3291–3297.
- [89] SIEK J, LEE LQ & LUMSDAINE A, 2014, *Boost graph library*, [Online], [Cited April 10th, 2014], Available from <http://www.boost.org/libs/graph/>.
- [90] SLATER PJ, 1995, *Locating dominating sets and locating-dominating sets*, pp. 1073–1079 in ALAVI Y & SCHWENK A (EDS), *Graph theory, combinatorics, and applications*, Proceedings of the 7th Quadrennial International Conference on the Theory and Applications of Graphs, John Wiley & Sons, Hoboken (NJ).
- [91] STEWART I, 1999, *Defending the Roman Empire!*, Scientific American, December, pp. 136–138.
- [92] SUMNER DP, 1990, *Critical concepts in domination*, Discrete Mathematics, **86**, pp. 33–46.
- [93] SUMNER DP & BLITCH P, 1983, *The discipline of a graph*, Journal of Combinatorial Theory, **34**, pp. 65–76.
- [94] TESCHNER U, 1997, *New results about the bondage number of a graph*, Discrete Mathematics, **171**, pp. 249–259.
- [95] THIERAUF T, 2000, *The computational complexity of equivalence and isomorphism problems*, Springer, Berlin.
- [96] UBUNTU EDITION: VERSION 12.04, 2012, *Canonical*, [Online], [Cited January 15th, 2013], Available from <http://www.canonical.com>.
- [97] VAN ROOIJ JMM & BODLAENDER HL, 2011, *Exact algorithms for dominating sets*, Discrete Applied Mathematics, **159(17)**, pp. 2147–216.

- [98] VAN ROOIJ JMM, NEDERLOF J & VAN DIJK TC, 2009, *Inclusion/exclusion meets measure and conquer: Exact algorithms for counting dominating sets*, pp. 554–565 in FIAT A & SANDERS P (EDS), *17th Annual European Symposium on Algorithms, Copenhagen*, Springer, New York (NY).
- [99] VIZING VG, 1963, *The cartesian product of graphs*, Vychislitel'nye Sistemy, **9**, pp. 30–43.
- [100] VIZING VG, 1965, *A bound on the external stability number of a graph*, Doklady Akademii Nauk SSSR, **164**, pp. 729–731.
- [101] VIZING VG, 1968, *Some unsolved problems in graph theory*, Russian Mathematical Surveys, **6(144)**, pp. 117–134.
- [102] WALIKAR HB & ACHARYA BD, 1979, *Domination critical graphs*, National Academy Science Letters, **2**, pp. 70–72.
- [103] WALIKAR HB, ACHARYA BD & SAMPATHKUMAR E, 1979, *Recent developments in the theory of domination of graphs*, MRI Lecture Notes in Mathematics, Mehta Research Institute, Allahabad.
- [104] WANG Y, 1996, *On the bondage number of a graph*, Discrete Mathematics, **159**, pp. 291–294.
- [105] WEST DB, 2001, *Introduction to graph theory*, 2nd Edition, Prentice Hall, Upper Saddle River (NJ).
- [106] WOLFRAM RESEARCH, 2005, *Mathematica Edition: Version 8.0*, [Online], [Cited May 30th, 2012], Available from <http://www.wolfram.com/mathematica>.

APPENDIX A

Properties of Floor and Ceiling Operations

Eight basic results related to various properties of the floor and ceiling operators are presented in this appendix for the purpose of easy referencing.

Since the *ceiling* of a real number a , denoted by $\lceil a \rceil$, is the smallest integer not exceeded by a , it follows trivially that $\lceil \bar{a} + \alpha \rceil = a + \lceil \alpha \rceil$ where $a = \bar{a} + \alpha$ with $\bar{a} \in \mathbb{Z}$ and $0 \leq \alpha < 1$.

Proposition A.1 *For any $a, b \in \mathbb{R}$, $\lceil a \rceil + \lceil b - a \rceil \geq \lceil b \rceil$.*

Proof: If a is an integer, then $\lceil a \rceil + \lceil b - a \rceil = a + \lceil b \rceil - a = \lceil b \rceil$, as desired.

Suppose, therefore, that $a = \bar{a} + \alpha$ and $b = \bar{b} + \beta$ with $\bar{a}, \bar{b} \in \mathbb{Z}$, $0 < \alpha < 1$ and $0 \leq \beta < 1$. Then

$$\begin{aligned} \lceil a \rceil + \lceil b - a \rceil &= \lceil \bar{a} + \alpha \rceil + \lceil \bar{b} + \beta - \bar{a} - \alpha \rceil \\ &= \lceil \alpha \rceil + \lceil \bar{b} + \beta - \alpha \rceil \\ &= \begin{cases} 1 + \lceil \bar{b} \rceil & \text{if } 0 < \alpha < \beta, \\ 1 + \lceil \bar{b} \rceil & \text{if } 0 \leq \beta \leq \alpha. \end{cases} \\ &\geq \lceil \bar{b} \rceil, \end{aligned}$$

as required. ■

The following results are useful in clarifying certain aspects of the proof of Theorem 7.18.

Proposition A.2 *For any $a \in \mathbb{Z}$ and $b \in \mathbb{R}$, $\lceil a + b \rceil = a + \lceil b \rceil$.*

Proof: Since the result is trivially true if $b \in \mathbb{Z}$, let $b = \bar{b} + \varepsilon$, with $\bar{b} \in \mathbb{Z}$ and $0 < \varepsilon < 1$. Then

$$\begin{aligned} \lceil a + b \rceil &= \lceil a + \bar{b} + \varepsilon \rceil \\ &= a + \bar{b} + 1 \\ &= a + \lceil \bar{b} + 1 \rceil \\ &= a + \lceil \bar{b} + \varepsilon \rceil \\ &= a + \lceil b \rceil, \end{aligned}$$

as desired. ■

The following result establishes the relationship between the ceiling of a non-integer and the ceiling of its negation.

Proposition A.3 For any $a \in \mathbb{R} \setminus \mathbb{Z}$, $\lceil -a \rceil = -\lfloor a \rfloor - 1$.

Proof: Let $a = \bar{a} + \varepsilon$, with $\bar{a} \in \mathbb{Z}$ and $0 < \varepsilon < 1$. Then

$$\begin{aligned} -\lceil -a \rceil &= -\lceil -\bar{a} - \varepsilon \rceil \\ &= -(-\bar{a}) \\ &= \lceil \bar{a} + 1 - 1 \rceil \\ &= \lceil \bar{a} + \varepsilon \rceil - 1 \\ &= \lceil a \rceil - 1, \end{aligned}$$

as desired. ■

The following result follows immediately from the proof of Proposition A.3 and its proof follows an identical progression to that of Proposition A.3 with the exception that now $\varepsilon = 0$.

Corollary A.1 For any $a \in \mathbb{Z}$, $\lceil -a \rceil = -\lfloor a \rfloor$. ■

The next result holds by Propositions A.2 and A.3.

Proposition A.4 For any $a \in \mathbb{Z}$ and $b \in \mathbb{R} \setminus \mathbb{Z}$, $a - \lceil b \rceil = \lceil a - b \rceil - 1$.

Proof: Using Propositions A.2 and A.3, it follows that

$$\begin{aligned} a - \lceil b \rceil &= -(-a + \lceil b \rceil) \\ &= -(\lceil -a + b \rceil) \\ &= \lceil -(-a + b) \rceil - 1 \\ &= \lceil a - b \rceil - 1, \end{aligned}$$

as desired. ■

Due to Corollary A.1 and Proposition A.4, the following result is possible.

Corollary A.2 For any $a \in \mathbb{Z}$ and $b \in \mathbb{R}$, $a - \lceil b \rceil \geq \lceil a - b \rceil - 1$. ■

The *floor* of a real number b , denoted by $\lfloor b \rfloor$, is the largest integer not exceeding b . The next result was originally posed as an exercise by Graham, Knuth and Patashnik [53, Exercise 3.12, pp. 96].

Proposition A.5 For any $a, b \in \mathbb{Z}$ with $b \neq 0$, $\lceil \frac{a}{b} \rceil = \lfloor \frac{a+b-1}{b} \rfloor$.

Proof: Let $a = kb - r$, where k is an integer and $0 \leq r < b$. Then

$$\begin{aligned} \lceil \frac{a}{b} \rceil &= \left\lceil \frac{kb - r}{b} \right\rceil \\ &= k + \left\lceil \frac{-r}{b} \right\rceil \\ &= k - \left\lfloor \frac{r}{b} \right\rfloor \\ &= k. \end{aligned}$$

Furthermore,

$$\begin{aligned}
 \left\lfloor \frac{a+b-1}{b} \right\rfloor &= \left\lfloor \frac{kb+r+b-1}{b} \right\rfloor \\
 &= \left\lfloor \frac{(k+1)b-r-1}{b} \right\rfloor \\
 &= k+1 + \left\lfloor \frac{-r-1}{b} \right\rfloor \\
 &= k+1 - \left\lceil \frac{r+1}{b} \right\rceil \\
 &= k+1-1 \\
 &= k.
 \end{aligned}$$

Since, $\lceil \frac{a}{b} \rceil = k = \lfloor \frac{a+b-1}{b} \rfloor$, the desired result follows. ■

The following result is similar to that of Proposition A.5.

Proposition A.6 For any $a, b \in \mathbb{Z}$ with $b \neq 0$, $\lfloor \frac{a}{b} \rfloor = \lceil \frac{a-b+1}{b} \rceil$.

Proof: Let $a = kb - r$, where k is a positive integer and $0 < r \leq b$. Then

$$\begin{aligned}
 \left\lfloor \frac{a}{b} \right\rfloor &= \left\lfloor \frac{kb-r}{b} \right\rfloor \\
 &= k + \left\lfloor \frac{-r}{b} \right\rfloor \\
 &= k - \left\lceil \frac{r}{b} \right\rceil \\
 &= k-1.
 \end{aligned}$$

Furthermore,

$$\begin{aligned}
 \left\lceil \frac{a-b+1}{b} \right\rceil &= \left\lceil \frac{kb+r-b+1}{b} \right\rceil \\
 &= \left\lceil \frac{(k-1)b-r+1}{b} \right\rceil \\
 &= k-1 + \left\lceil \frac{-r+1}{b} \right\rceil \\
 &= k-1 - \left\lfloor \frac{r-1}{b} \right\rfloor \\
 &= k-1+0 \\
 &= k-1.
 \end{aligned}$$

Since, $\lfloor \frac{a}{b} \rfloor = k-1 = \lceil \frac{a-b+1}{b} \rceil$, the desired result follows. ■

The following result is the final result required in the proof of Theorem 7.18.

Proposition A.7 For any $a, b, c \in \mathbb{N}$, $\lceil \frac{a}{b} \rceil + \lceil c - \frac{a}{b} \rceil = c$.

Proof: Let $a = kb - r$, where k is a positive integer and $0 \leq r < b$. Then

$$\begin{aligned} \left\lceil \frac{a}{b} \right\rceil &= \left\lceil \frac{kb + r}{b} \right\rceil \\ &= k + \left\lceil \frac{r}{b} \right\rceil \\ &= k + 1. \end{aligned}$$

Furthermore,

$$\begin{aligned} \left\lceil c - \frac{a}{b} \right\rceil &= \left\lceil c - \frac{kb + r}{b} \right\rceil \\ &= c + \left\lceil \frac{-(kb + r)}{b} \right\rceil \\ &= c - k + \left\lceil \frac{-r}{b} \right\rceil \\ &= c - k - \left\lceil \frac{r}{b} \right\rceil \\ &= c - k - 1. \end{aligned}$$

Since, $\left\lceil \frac{a}{b} \right\rceil + \left\lceil c - \frac{a}{b} \right\rceil = (k + 1) + (c - k - 1) = c$, the desired result follows. ■

The following final result of this appendix is required in the proof of Theorem 7.17.

Proposition A.8 For any $a, b \in \mathbb{N}$, $\left\lceil \frac{a}{b} \right\rceil + \left\lceil \frac{1-a}{b} \right\rceil \leq 1$.

Proof: Let $a = kb - r$, where k is an integer and $0 \leq r < b$. Then

$$\begin{aligned} \left\lceil \frac{a}{b} \right\rceil &= \left\lceil \frac{kb + r}{b} \right\rceil \\ &= k + \left\lceil \frac{r}{b} \right\rceil \\ &= k + 1. \end{aligned}$$

Furthermore,

$$\begin{aligned} \left\lceil \frac{1-a}{b} \right\rceil &= \left\lceil \frac{1 - (kb + r)}{b} \right\rceil \\ &= -k + \left\lceil \frac{1-r}{b} \right\rceil \\ &= -k - \left\lceil \frac{r-1}{b} \right\rceil \\ &= \begin{cases} -k - 1 & \text{if } 0 \leq r < 1, \\ -k & \text{otherwise.} \end{cases} \end{aligned}$$

Since,


$$\left\lceil \frac{a}{b} \right\rceil + \left\lceil \frac{1-a}{b} \right\rceil = \begin{cases} 0 & \text{if } 0 \leq r < 1, \\ 1 & \text{otherwise,} \end{cases}$$




the desired result follows. ■











APPENDIX B

Repositories of critical and stable graphs

This appendix contains graphical illustrations of the nonempty classes of critical graphs and stable graphs of orders not exceeding 6.

\mathcal{Q}_2^q	q -Critical graphs of order 2
\mathcal{Q}_2^1	

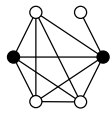
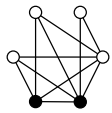
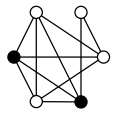
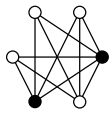
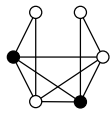
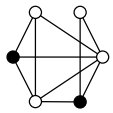
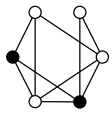
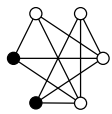
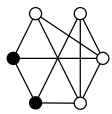
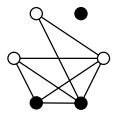
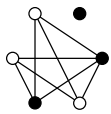
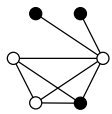
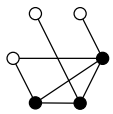
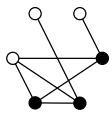
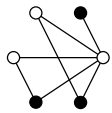
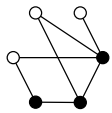
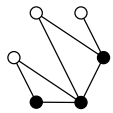
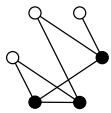
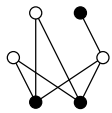
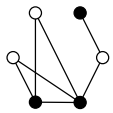
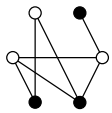
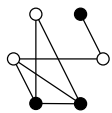
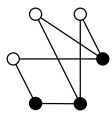
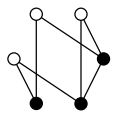
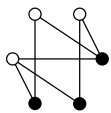
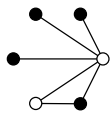
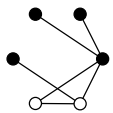
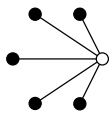
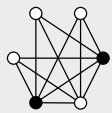
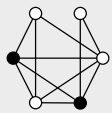
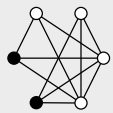
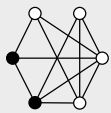
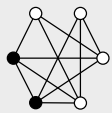
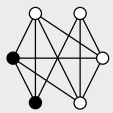
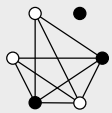
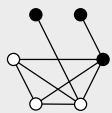
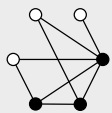
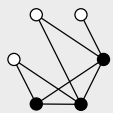
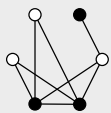
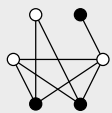
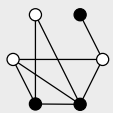
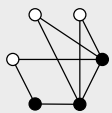
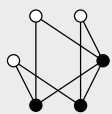
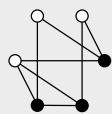
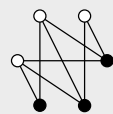
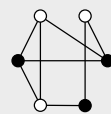
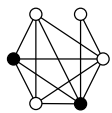
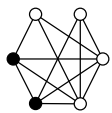
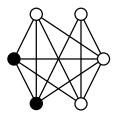
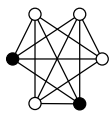
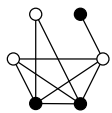
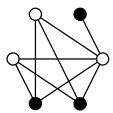
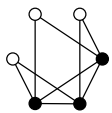
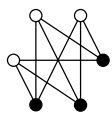
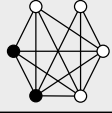
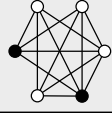
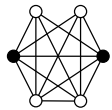
\mathcal{Q}_3^q	q -Critical graphs of order 3
\mathcal{Q}_3^1	 
\mathcal{Q}_3^2	

\mathcal{Q}_4^q	q -Critical graphs of order 4
\mathcal{Q}_4^1	   
\mathcal{Q}_4^2	 
\mathcal{Q}_4^3	  
\mathcal{Q}_4^4	

Q_5^q	q -Critical graphs of order 5							
Q_5^1								
Q_5^2								
Q_5^3								
Q_5^4								
Q_5^5								
Q_5^6								

Q_6^q	q -Critical graphs of order 6						
Q_6^1							
Q_6^2							

\mathcal{Q}_6^q	q -Critical graphs of order 6 (continued)						
\mathcal{Q}_6^3							
\mathcal{Q}_6^4							

\mathcal{Q}_6^q	q -Critical graphs of order 6 (continued)						
\mathcal{Q}_6^5							
							
							
							
\mathcal{Q}_6^6							
							
							
\mathcal{Q}_6^7							
							
\mathcal{Q}_6^8							
\mathcal{Q}_6^9							

\mathcal{S}_2^p p -Stable graphs of order 2







\mathcal{S}_2^0	
-------------------	---

\mathcal{S}_3^p p -Stable graphs of order 3

\mathcal{S}_3^0	 
-------------------	---

\mathcal{S}_3^1	
-------------------	---


















\mathcal{S}_4^p p -Stable graphs of order 4


















\mathcal{S}_4^0	     
-------------------	---


\mathcal{S}_4^1	  
-------------------	---

\mathcal{S}_4^2	
-------------------	---

\mathcal{S}_5^p p -Stable graphs of order 5

\mathcal{S}_5^0	       
	       
	

\mathcal{S}_5^1	       
	       
	

\mathcal{S}_5^2	  
-------------------	---

\mathcal{S}_5^3	
-------------------	---

\mathcal{S}_6^p p -Stable graphs of order 6	

\mathcal{S}_6^p p -Stable graphs of order 6 (continued)							
\mathcal{S}_6^1							
\mathcal{S}_6^2							
\mathcal{S}_6^3							
\mathcal{S}_6^4							

APPENDIX C

Contents of the accompanying compact disc

A brief description of the contents of the compact disc included with the dissertation is given in this appendix. The compact disc contains computer implementations of all the algorithms described and analysed in the dissertation. Furthermore, the disc contains the edge-removal metagraphs of the complete graphs \mathcal{K}_n of orders $n \in \{3, \dots, 9\}$ described in §7.1, as well as the nonempty classes of critical and stable graphs of orders not exceeding 9.

The branch-and-bound algorithm of §5.1.1 and the branch-and-reduce algorithm of §5.1.2 were implemented in Wolfram's **Mathematica** [106], the binary programming model of §5.1.5 was implemented in the software suite **CPLEX** [37], and the remaining algorithms were implemented in **C++**. Details on the compilation and usage of the computer code are provided on the compact disc. The compact disc contains the following four directories:

Algorithms for the secure domination number. This directory contains four subdirectories, namely “Linear tree algorithm”, “Branch-and-reduce algorithm”, “Branch-and-bound algorithm” and “Binary programming formulation,” which contain the relevant algorithmic implementations for computing the secure domination number of a graph.

Algorithms for generating critical and stable graphs. This directory contains three subdirectories, namely “Edge-removal metagraph”, “Critical graphs” and “Stable graphs,” which contain the relevant algorithmic implementations for computing the edge-removal metagraph of a graph and the various classes of critical and stable graph classes presented in §7.2–§7.3.

Repository. Text files are provided in this directory containing adjacency matrices of the edge-removal metagraphs of complete graphs, as well as the non-empty classes critical and stable graphs of orders not exceeding 9.

Boost. This directory contains the current boost library [89] used in the algorithmic implementations in this dissertation.

Dissertation. This directory contains an electronic copy of the dissertation.